Parikh Automata and Monadic Second-Order Logics with Linear Cardinality Constraints

Felix Klaedtke[1] and Harald Rueß[2]

[1] Institut für Informatik
Albert-Ludwigs-Universität
Georges-Köhler-Allee 52
79085 Freiburg i. Br., Germany
Phone: +49 (761) 203-8245
Fax: +49 (761) 203-8242
`klaedtke@informatik.uni-freiburg.de`
contact author

[2] Computer Science Laboratory
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025, USA
Phone: +01 (650) 859-5710
Fax: +01 (650) 859-2844
`ruess@csl.sri.com`

Abstract. We contribute to the classical automaton-logic connection. On the one hand, we generalize the acceptance condition of word and tree automata by additionally requiring that the input satisfies certain arithmetic properties. On the other hand, we define extensions of monadic second-order logics on words and trees that allow us to compare cardinalities of sets. Although these extensions are undecidable we identify useful decidable fragments.

Keywords: WS1S and WS2S, extensions of finite word and tree automata, commutative image, semi-linear sets, Presburger arithmetic.

1 Introduction

Monadic second-order logics on words and trees were introduced over forty years ago to solve certain decision problems. For example, the decidability of the (weak) monadic second-order logic of one successor (W)S1S was proved by a translation of formulas to word automata [6, 7, 12, 31], and the decidability of the (weak) monadic second-order logic on binary trees (W)S2S was shown by a translation to tree automata [9, 24, 29]. Despite the staggering, non-elementary, worst-case complexity of these decision procedures [21, 28], it was through the Mona tool [10, 14] that it became possible to make effective use of automata-based decision procedures for WS1S and WS2S for modeling and reasoning about computation systems such as families of hardware circuits [2], imperative programs with pointers [11], and finite transition systems [14, 19]. Moreover, this automata-theoretic decision procedure for WS1S is also used to decide well-defined fragments of higher-order logic [1, 22].

However, many interesting problems, such as the *Byzantine Generals* problem [20] of reaching distributed consensus in the presence of unreliable messengers and treacherous generals, fall outside the scope of these logics, since they require reasoning about the combination of (finite) sets and cardinality constraints on these sets. In order to support this kind of reasoning we extend

WS1S and WS2S with cardinality constraints of the form

$$c_0 + \sum_{i=1}^{r} c_i|X_i| < d_0 + \sum_{j=1}^{s} d_j|Y_j|\,,$$

where the c_is and d_js are natural numbers, and $|X|$ denotes the cardinality of the set X. Here, we restrict ourselves to weak interpretations, i.e., monadic second-order (MSO) variables are interpreted only over finite sets. The extension of WS1S (WS2S) with cardinality constraints is denoted by WS1S$^{\mathrm{card}}$ (WS2S$^{\mathrm{card}}$). Our main results are:

(i) WS1S$^{\mathrm{card}}$ and WS2S$^{\mathrm{card}}$ are undecidable. In fact, already the fragment of WS1S$^{\mathrm{card}}$ consisting of sentences of the form $\forall\overline{X}\exists\overline{Y}\varphi$ is undecidable, where $\overline{X}$ and $\overline{Y}$ are vectors of MSO variables ranging over finite sets of natural numbers, and all quantifiers in φ are first-order, i.e., they range over natural numbers.

(ii) The fragment of WS1S$^{\mathrm{card}}$ consisting of sentences of the form

$$Q_1x_1 \dots Q_mx_mQ\overline{Y}\varphi$$

is decidable, where $Q_1, \dots, Q_m \in \{\exists, \forall\}$ are quantifiers over natural numbers, $Q \in \{\exists, \forall\}$ quantifies the MSO variables in $\overline{Y}$, and all MSO variables occurring in a cardinality constraint in φ are bounded by one of the quantifiers for $\overline{Y}$.

(iii) We show decidability of a fragment of WS2S$^{\mathrm{card}}$, analogously to the fragment above for WS1S$^{\mathrm{card}}$.

These results frame the landscape of decidability versus undecidability of MSO logics with cardinality constraints. Despite the rather drastic restrictions on the decidable fragment of WS1S$^{\mathrm{card}}$ in (ii), we can show that this fragment is expressive enough to encode oral message algorithms such as *OM(1)*, together with fault assumptions and the required agreement and validity properties [17], thereby opening up the possibility of automatic verification of these algorithms. In contrast, formal construction of proofs for oral message algorithm in various settings and under different error scenarios, such as the one by Rushby [25], are cumbersome and rather involved, since they usually rely on interactive guidance and prover libraries about cardinalities of finite sets.

The main device for proving the results (i) and (ii) are word automata with a generalized acceptance condition based on Parikh's [23] *commutative image* of words. This new kind of device which we name *Parikh automata* works on an extended alphabet, where we attach vectors of natural numbers to each symbol of the input alphabet. A word over the input alphabet is accepted if an extension of the word is accepted in the traditional sense of automata, and, additionally, arithmetic constraints are satisfied, where the extended word is projected via a monoid homomorphism to a vector of natural numbers. Parikh automata are closed under union and intersection but not under complement. The emptiness problem for Parikh automata is shown to be decidable using Parikh's characterization of the commutative image of context-free languages as the finite union of the images of linear polynomials [23].

The result (i) follows from the facts that $\mathrm{WS1S}^{\mathrm{card}}$ is expressive enough to describe Parikh automata and the undecidablity of the universality problem for Parikh automata. The proof for (ii) is more difficult because of the prefix of the quantified FO variables. The decision procedure for (ii) consists of two constructions. For an instance $Q_1 x_1 \ldots Q_m x_m Q\overline{Y}\varphi$ of the decision problem, we first construct a Parikh automaton that recognizes the language of the models of the $\mathrm{WS1S}^{\mathrm{card}}$ formula $\varphi(x_1, \ldots, x_m, \overline{Y})$. This construction step is analogously as in the automata-based decision for WS1S. Second, from this Parikh automaton we construct a Presburger arithmetic formula that is true iff the $\mathrm{WS1S}^{\mathrm{card}}$ sentence $Q_1 x_1 \ldots Q_m x_m Q\overline{Y}\varphi$ is true.

The classical automaton-logic connection extends in the following way to Parikh automata and $\mathrm{WS1S}^{\mathrm{card}}$. On the one hand, $\mathrm{WS1S}^{\mathrm{card}}$ is expressive enough to describe Parikh automata by formulas of the form $\exists\overline{Y}\varphi$, where $\overline{Y}$ are MSO variables and all quantifiers in φ are first-order. On the other hand, a formula of the form $\exists\overline{Y}\varphi$, where all MSO variables occurring in a cardinality constraint in φ are either free or bounded by one of the quantifiers for the MSO variables $\overline{Y}$, can be translated to a Parikh automaton recognizing the language of all its models. This connection extends further to Parikh tree automata and $\mathrm{WS2S}^{\mathrm{card}}$.

The rest of this paper is structured as follows. In §2 we introduce Parikh word automata and examine closure properties, the emptiness and universality problem, and compare Parikh word automata with other automata models found in the literature. Then, in §3 we define $\mathrm{WS1S}^{\mathrm{card}}$, and prove the results (i) and (ii). In §4 we introduce Parikh tree automata and carry over the results for $\mathrm{WS1S}^{\mathrm{card}}$ to $\mathrm{WS2S}^{\mathrm{card}}$ to establish (iii). Finally, in §5 we draw conclusions.

Note to the reader: detailed technical expositions that are only sketched in the main text or omitted altogether can be found in the appendix.

2 Parikh Automata

We generalize the acceptance condition of automata. Instead of only requiring that there is a run ending in a final state, we also require that an input word satisfies an arithmetic property, where the input word is associated with a vector of natural numbers.

Given a finite, linearly ordered alphabet $\Sigma = \{b_1, \ldots, b_n\}$, elements of the free monoid Σ^*, so-called words, are mapped via Parikh's [23] *commutative image* $\Phi : \Sigma^* \to \mathbb{N}^n$ to vectors of natural numbers. The commutative image is defined by

$$\Phi(b_i) := \overline{e}_i \qquad \text{and} \qquad \Phi(uv) := \Phi(u) + \Phi(v)\,,$$

where $\overline{e}_i$ is the unit vector where the ith coordinate is 1. Intuitively, the ith position of $\Phi(w)$ counts how often the symbol b_i occurs in $w \in \Sigma^*$. We extend the concept of Parikh's commutative image by considering the Cartesian product of Σ and a finite set D of vectors of natural numbers.

Definition 1. *Let Γ be an alphabet of the form $\Sigma \times D$ where $D \subseteq \mathbb{N}^N$, for some $N \in \mathbb{N}$. We define the* ***projection*** $\Psi : \Gamma^* \to \Sigma^*$ *and the* ***Parikh map*** $\Phi : \Gamma^* \to \mathbb{N}^N$ *as monoid homomorphisms.*

(i) $\Psi(b, \overline{d}) := b$ *and* $\Psi(uv) := \Psi(u)\Psi(v)$.
(ii) $\Phi(b, \overline{d}) := \overline{d}$ *and* $\Phi(uv) := \Phi(u) + \Phi(v)$.

We constrain a language by arithmetic properties expressed in Presburger arithmetic.

Definition 2. *For a language $L \subseteq (\Sigma \times D)^*$, where $D \subseteq \mathbb{N}^n$, the structure $\mathfrak{N} := (\mathbb{N}, +, <, 0, 1)$, and a Presburger arithmetic formula $\varphi(x_1, \ldots, x_n)$, let*

$$L\restriction_\varphi := \{\Psi(w) \mid w \in L \text{ and } \mathfrak{N} \models \varphi[\Phi(w)/\overline{x}]\},$$

be the ***restriction*** *of L with respect to φ.*

As usual, $\varphi(x_1, \ldots, x_n)$ denotes that at most the variables $x_1, \ldots, x_n$ occur free in the formula φ, and $\varphi[t_1/x_1, \ldots, t_n/x_n]$ denotes the simultaneous substitution of each variable x_i by the term t_i. Whenever n is clear from the context, we often just write $\varphi[\overline{t}/\overline{x}]$. We also use syntactic sugar such as writing $2x + 2$ for the formula $x + x + 1 + 1$.

Now, we define Parikh automata as automata operating on words over an alphabet $\Sigma \times D$. In addition to the acceptance condition of automata the inputs are also required to satisfy certain arithmetic properties.

Definition 3. *A* ***Parikh automaton*** *of dimension $N \geq 0$ is a pair $(\mathcal{A}, \varphi)$, where*

(i) $\mathcal{A}$ is a finite word automaton with the alphabet $\Sigma \times D$, where $D \subseteq \mathbb{N}^N$, and
(ii) $\varphi(z_1, \ldots, z_N)$ is a Presburger arithmetic formula.

$(\mathcal{A}, \varphi)$ ***recognizes*** *the language $L(\mathcal{A}, \varphi) := L(\mathcal{A})\restriction_\varphi$, where $L(\mathcal{A})$ is the language recognized by $\mathcal{A}$.*

A Parikh automaton $(\mathcal{A}, \varphi)$ is ***deterministic*** *if for the transition function δ of $\mathcal{A}$ it holds that for every state q and for every $(b, \overline{d}) \in \Sigma \times D$, $|\delta(q, (b, \overline{d}))| \leq 1$, and if $|\delta(q, (b, \overline{d}))| = 1$ then $|\delta(q, (b, \overline{d}'))| = 0$, for every $\overline{d} \neq \overline{d}'$.*

For example, the deterministic Parikh automaton $(\mathcal{A}, z_1 = z_2)$ of dimension 2, where $\mathcal{A}$ is depicted in Figure 1, recognizes the context-sensitive language $\{a^{m+n}b^m c^n \mid m, n > 0\}$ which is not context-free.

Intuitively, a Parikh automaton recognizes a word that satisfies both a "local" and a "global" property. For example, the local property for the Parikh automaton $(\mathcal{A}, z_1 = z_2)$ is that every word must contain at least one b and one c, the as occur before the bs, and the bs occur before the cs. The global property is that the sum of the occurrences of bs and cs is equal to the occurrences of as.

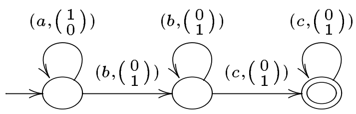

Fig. 1. An example of an underlying automaton of a Parikh automaton.

Parikh automata are strictly more expressive than regular languages constrained by arithmetic properties over the commutative image. It is easy to define a deterministic Parikh automaton that recognizes the language $L := \{a^m b^m a^n b^n \mid m, n \geq 1\}$. But there do not exist an automaton $\mathcal{A}$ with the alphabet $\{a, b\}$ and a Presburger arithmetic formula $\varphi(z_1, z_2)$ such that $w \in L$ iff $w \in L(\mathcal{A})$ and $\mathfrak{N} \models \varphi[k/z_1, \ell/z_2]$, where (k, ℓ) is the commutative image of w.

2.1 Some Properties about Parikh Automata

In this subsection, we investigate some closure properties of Parikh automata, and the emptiness and universality problem.

Property 4. *(1) Parikh automata are closed under union, intersection, homomorphisms, inverse homomorphisms, concatenations, and left and right quotients, but not under complement.*
(2) Deterministic Parikh automata are closed under union, intersection, complement, and inverse homomorphisms, but not under homomorphisms.

The decidability of the emptiness problem relies on Parikh's fundamental result [23] which states that the commutative image of a context-free language L is semi-linear, i.e., the commutative image of L is the union of the images of finitely many linear polynomials.

Lemma 5. *Let Γ be an alphabet of the form $\Sigma \times D$ where $D \subseteq \mathbb{N}^N$, for some $N \geq 0$. For every context-free language $L \subseteq \Gamma^*$, there exist linear polynomials $q_1, \ldots, q_m$ such that*

$$\Phi(L) = \bigcup_{1 \leq i \leq m} \{q_i(x_1, \ldots, x_r) \mid x_1, \ldots, x_r \in \mathbb{N}\},$$

where Φ is the Parikh map of Γ. Moreover, the polynomials $q_1, \ldots, q_m$ are effectively constructible.

Proof. Assume that $L \subseteq \{(b_1, \overline{v}_1), \ldots, (b_s, \overline{v}_s)\}^*$. By Parikh's theorem [23] we can construct linear polynomials $p_1, \ldots, p_m : \mathbb{N}^r \to \mathbb{N}^s$, for some $r \geq 0$, such that the commutative image of L is the union of the images of these polynomials. Recall that for each $1 \leq i \leq s$, the commutative image of a word $w \in \Gamma^*$ counts in its ith coordinate the number of occurences of the symbol $(b_i, \overline{v}_i)$.

Assume that the polynomial p_i, for $1 \leq i \leq m$ is of the form

$$p_i(x_1, \ldots, x_r) = \overline{d}_{i0} + \overline{d}_{i1} x_1 + \cdots + \overline{d}_{ir} x_r,$$

with $\overline{d}_{ij} = (d_{ij}^1, \ldots, d_{ij}^s)$, for $0 \leq j \leq r$. For $1 \leq k \leq s$, let $p_i^k : \mathbb{N}^r \to \mathbb{N}$ denote the polynomial $d_{i0}^k + d_{i1}^k x_1 + \cdots + d_{ir}^k x_r$. For $1 \leq i \leq m$, we define the polynomial $q_i : \mathbb{N}^r \to \mathbb{N}^N$ as

$$q_i(x_1, \ldots, x_r) := p_i^1(x_1, \ldots, x_r)\overline{v}_1 + \cdots + p_i^s(x_1, \ldots, x_r)\overline{v}_s \,. \qquad \dashv$$

For a Parikh automaton $(\mathcal{A}, \varphi)$ we can construct, by Lemma 5, polynomials $q_1, \ldots, q_m$ such that

$$\Phi(L(\mathcal{A})) = \bigcup_{1 \leq i \leq m} \{q_i(x_1, \ldots, x_r) \mid x_1, \ldots, x_r \in \mathbb{N}\} \,.$$

It follows that

$$L(\mathcal{A}, \varphi) \neq \emptyset \qquad \text{iff} \qquad \mathfrak{N} \models \exists x_1 \ldots \exists x_r (\bigvee_{1 \leq i \leq m} \varphi[q_i(x_1, \ldots, x_r)/\overline{z}]) \,.$$

The decidability of the emptiness problem follows immediately from the decidability of Presburger arithmetic.

Property 6. *The emptiness problem for Parikh automata is decidable.*

The undecidability of the universality problem for Parikh automata can be shown by reducing the word problem for Turing machines to it.

Property 7. *The universality problem for Parikh automata is undecidable.*

2.2 Comparison of Parikh Automata to Other Automata Models

There are many extensions and variations of automata over words in the literature. In this subsection, we compare some of them to Parikh automata.

Parikh automata can be seen as automata extended with counters, where a vector of natural numbers attached to a symbol is interpreted as an increment of the counters. In contrast to other counter automata models in the literature, e.g. [8, 13, 15], we do not restrict the applicability of transitions in a run by additional guards on the values of the counters. Instead, a Parikh automaton constrains the language of an automaton over the extended alphabet with a Presburger arithmetic formula. It turns out that Parikh word automata are as expressive as Ibarra's *reversal-bounded multicounter machines* [15]. Ibarra's [15] reversal-bounded multicounter machines are two-way automata augmented with counters, where the reversals of the movement of the input-head, and the reversals of decrement and increment of the counters are bounded.

Property 8. *Reversal-bounded multicounter machines and Parikh automata are equivalent in the sense that both devices describe the same set of languages.*

Despite this equivalence, Parikh automata have several advantages. For the simple, mathematical definition of Parikh automata, arguably, proofs are usually less technical. More importantly, Parikh automata can easily be generalized in several directions. For example, we can define Parikh pushdown automata by using pushdown automata instead of finite word automata in Definition 3. Since Parikh's theorem holds for context-free languages, the emptiness problem remains decidable for Parikh pushdown automata. A second extension is to replace Presburger arithmetic in Definition 3 by some decidable logic, such as Büchi arithmetic [5, 6]. A third extension is to consider trees instead of words, as we will do in §4.

The idea of extending the acceptance condition of automata by linear arithmetical constraints was also used in the definition of *constrainted queue-content decision diagrams* (CQDDs) [4]. CQDDs are word automata of some very special form together with linear arithmetical constraints on the number of times certain transitions in these automata are taken in a run. In our proposed framework of restricting the languages by formulas in Presburger arithmetic, a CQDD of dimension $n \geq 1$ over the alphabet Σ can be viewed as a set $\{(\mathsf{A}_1, \psi_i), \dots, (\mathsf{A}_m, \psi_m)\}$, where the A_is are n-tuples $(\mathcal{A}_{i1}, \dots, \mathcal{A}_{in})$ of word automata over an alphabet $\Sigma \times \{0,1\}^r$, for some $r \geq 0$ that is large enough, and the ψ_is are Presburger arithmetic formulas. Each automaton $\mathcal{A}_{ij}$ has to be of a special form, namely $\mathcal{A}_{ij}$ is a *deterministic restricted simple automaton* [4]. Let Φ be the Parikh map and Ψ be the projection for $(\Sigma \times \{0,1\}^r)^*$. The language of the CQDD is the set

$$\bigcup_{1 \leq i \leq m} \{ (\Psi(w_1), \dots, \Psi(w_n)) \mid (w_1, \dots, w_n) \in L(\mathcal{A}_{i1}) \times \dots \times L(\mathcal{A}_{in}) \text{ and } \\ \mathfrak{N} \models \psi_i[\Phi(w_1) + \dots + \Phi(w_n)/\overline{z}] \} .$$

CQDDs were used in [4] to describe sets of configurations of FIFO-channel systems. Intuitively, a tuple $(u_1, \dots, u_n) \in (\Sigma^*)^n$ in the language of a CQDD corresponds to a configuration of a FIFO-channel system, where the word u_i is the content of the ith channel. Although CQDDs can describe non-regular languages, like for example $\{a^m b^n a^m b^n \mid m, n \geq 1\}$, there are regular languages that cannot be described by CQDDs because of the restrictions on the word automata $\mathcal{A}_{ij}$. For example, the language given by the regular expression $(ab^*a)^*$ cannot be described by a CQDD. It holds that CQDDs of dimension 1 are strictly less expressive than deterministic Parikh automata.

Another extension of word automata are *multiplicity automata* introduced by Schützenberger [26]. A multiplicity automaton is essentially a nondeterministic word automaton where each transition is augmented with a weight. For every path the automaton assigns a weight which is the product of the weights of the transitions in the path. A multiplicity automaton essentially outputs the sum of weights of paths consistent with the input string. Although multiplicity automata also extend word automata they are fundamentally different to Parikh automata. For example, the language $\{a^{n+1} b a^{n+1} \mid n \geq 0\}$ cannot be recognized by a multiplicity automaton [26] but it is straightforward to define a deterministic Parikh automaton that recognizes this language.

3 WS1S with Cardinality Constraints

We extend WS1S in order to compare cardinalities of sets. We call this extension WS1S$^{\mathrm{card}}$. The classical automata-logic connection of word automata and WS1S [6, 12, 31] extends to Parikh automata and a fragment of WS1S$^{\mathrm{card}}$. Although WS1S$^{\mathrm{card}}$ is undecidable, we can use Parikh automata to show decidability of a fragment of it that nontrivially extends WS1S.

3.1 The Weak Monadic Second-Order Logic of One Successor

The atomic formulas of WS1S are membership Xx, and the successor relation $\mathrm{succ}(x, y)$, where x, y are first-order (FO) variables, and X is a monadic second-order (MSO) variable. In the following, we adopt the following notation. Lower case letters $x, y, \ldots$ denote FO variables and upper case letters $X, Y, \ldots$ denote MSO variables. Moreover, $\alpha, \beta, \ldots$ range over FO and MSO variables. Formulas are built from the atomic formulas and the connectives $\neg$ and $\vee$, and the existential quantifier $\exists$ for FO and MSO variables. We also use the connectives $\wedge$, $\rightarrow$ and $\leftrightarrow$, and the universal quantifiers $\forall$ for FO and MSO variables. We use the standard conventions for omitting parenthesis. A sentence is a formula with no free variables.

Formulas are interpreted over the natural numbers with the successor relation, i.e., the structure $(\mathbb{N}, \mathrm{succ})$. A (weak) interpretation I maps FO variables to natural numbers and MSO variables are mapped to *finite* subsets of $\mathbb{N}$. The truth value of a formula φ in $(\mathbb{N}, \mathrm{succ})$ with respect to an interpretation I, in symbols $(\mathbb{N}, \mathrm{succ}), I \models \varphi$, is defined as expected. Note that existential quantification for MSO variables only ranges over *finite* subsets of $\mathbb{N}$. We write $(\mathbb{N}, \mathrm{succ}) \models \varphi$ if φ is a sentence.

We write $x \leq y$ for the formula

$$\forall Z\big(Zy \wedge \forall z \forall z'(Zz' \wedge \mathrm{succ}(z, z') \rightarrow Zz) \rightarrow Zx\big)\ ,$$

$x = y$ for $x \leq y \wedge y \leq x$, and $x < y$ for $\neg x = y \wedge x \leq y$. For $k \geq 0$, let $\psi_{\max}(y, X_1, \ldots, X_k)$ be the formula $\forall z(y < z \rightarrow \bigwedge_{1 \leq i \leq k} \neg X_i z) \wedge \bigvee_{1 \leq i \leq k} X_i y$, i.e., y is the maximum of $X_1, \ldots, X_k$. And let $\psi_{\mathrm{part}}(y, X_1, \ldots, X_k)$ be the formula stating that $X_1, \ldots, X_k$ is a partition of $\{0, \ldots, y\}$, i.e., $\psi_{\mathrm{part}}(y, X_1, \ldots, X_k)$ is the formula $\forall x(x \leq y \leftrightarrow \bigvee_{1 \leq i \leq k} X_i x) \wedge \neg \exists x\big(\bigvee_{1 \leq i < j \leq k}(X_i x \wedge X_j x)\big)$.

The decidability of WS1S is based on a construction that builds for a formula an automaton recognizing the words that represent the interpretations making the formula true. A word $w = b_1 \ldots b_n \in (\{0,1\}^k)^*$ determines the interpretation I_w for variables $\alpha_1, \ldots, \alpha_k$, where for all $1 \leq j \leq k$,

- $i = I_w(\alpha_j)$ iff $\chi_j(b_i) = 1$ and $\chi_j(b_{i'}) = 0$ for all $i' \neq i$, whenever α_i is a FO variable, and
- $i \in I_w(\alpha_j)$ iff $\chi_j(b_i) = 1$, whenever α_j is a MSO variable,

where χ_j projects a vector in $\{0,1\}^k$ to its jth coordinate. We extend χ_j homomorphically to words. Note that we have implicitly assumed that the variables

$\alpha_1, \ldots, \alpha_k$ are ordered in the sense that the interpretation I_w of the variable α_j is determined by the jth projection of $b_1 \ldots b_n$, i.e. $\chi_j(b_1 \ldots b_n)$. In the following, we write χ_{α_j} for χ_j. For a formula $\varphi(\alpha_1, \ldots, \alpha_k)$, we define

$$L(\varphi) := \{w \in (\{0,1\}^k)^* \mid (\mathbb{N}, \mathrm{succ}), I_w \models \varphi\}\,.$$

Later, we shall need the following fact that is due to Büchi, Elgot, and Trakhtenbrot [6, 12, 31]. For more details, see, e.g., [30].

Fact 9. $L(\varphi)$ is regular for every WS1S formula φ. Moreover, we can effectively build an automaton recognizing $L(\varphi)$.

WS1S$^{\mathrm{card}}$ has in addition to the atomic formulas of WS1S the atomic formulas of the form

$$c + |X_1| + \cdots + |X_r| < d + |Y_1| + \cdots + |Y_s|\,, \tag{$*$}$$

where $c, d \in \mathbb{N}$ are constants. The truth value of the formula $(*)$ with respect to an interpretation I is defined as

$$(\mathbb{N}, \mathrm{succ}), I \models c + |X_1| + \cdots + |X_r| < d + |Y_1| + \cdots + |Y_s| \quad \text{iff} \quad c + \sum_{1 \le i \le r} |I(X_i)| < d + \sum_{1 \le i \le s} |I(Y_i)|\,.$$

We will write formulas like $-2|X| = 3|Y| - 1$ which can be transformed to a Boolean combination of formulas in C by standard arithmetic. Moreover, we will also use the summation symbol $\sum$ for a shorter representation of formulas in C. We define the following sets of formulas.

Definition 10. *(i) Let C be the set of atomic formulas of the form $(*)$ and their negations.*
(ii) For $Q \in \{\exists, \forall\}$, let F_Q be the set of WS1S$^{\mathrm{card}}$ *formulas of the form*

$$QX_1 \ldots QX_n\varphi\,,$$

where all MSO variables occurring in a subformula in C of φ are either free or bounded by one of the quantifiers for $X_1, \ldots, X_n$.

3.2 Parikh Automata and WS1S$^{\mathrm{card}}$

The next theorem shows that the fragment $\mathsf{F}_\exists$ is expressive enough to describe Parikh automata. The **word model** of $b_1 \ldots b_n \in \Sigma^*$ is the structure $(\mathbb{N}, \mathrm{succ})$ extended with the family of predicates $(U_b)_{b \in \Sigma}$, where $i - 1 \in U_b$ iff $b_i = b$. Let $(\mathbb{N}, \mathrm{succ}, w)$ denote the word model of $w \in \Sigma^*$.

Theorem 11. *Let $(\mathcal{A}, \varphi)$ be a Parikh automaton with $L(\mathcal{A}, \varphi) \subseteq \Sigma^*$. There is a formula $\psi_{\mathcal{A},\varphi} \in \mathsf{F}_\exists$ with the free variables U_b for $b \in \Sigma$ such that*

$$w \in L(\mathcal{A}, \varphi) \qquad \textit{iff} \qquad (\mathbb{N}, \mathrm{succ}, w) \models \psi_{\mathcal{A},\varphi}\,,$$

for every word $w \in \Sigma^*$. Moreover, $\psi_{\mathcal{A},\varphi}$ is of the form $\exists X_1 \ldots \exists X_n \chi$, where all bounded variables in χ are FO variables.

Proof. Let $N \geq 0$ be the dimension of $(\mathcal{A}, \varphi)$, and let $\mathcal{A} = (Q, \Sigma \times D, \delta, q_{\mathrm{I}}, F)$. Without loss of generality, we assume that $Q = \{1, \ldots, r\}$ and $\Sigma = \{1, \ldots, s\}$, for some $r, s \geq 1$. Let K be the maximal natural number occurring in a vector in D, i.e. $K := \max(\{0\} \cup \bigcup_{(d_1,\ldots,d_N)\in D}\{d_1, \ldots, d_N\})$. Presburger arithmetic admits quantifier elimination, i.e., we can construct for every Presburger arithmetic formula an equivalent quantifier-free formula over the first-order language $<$, $+$, 0, 1, and $\cdot \equiv \cdot \bmod n$, for every $n \geq 2$, where $\cdot \equiv \cdot \bmod n$ expresses congruence modulo the fixed divisor n. Therefore, we can assume that φ is such a quantifier free formula.

For a word $w = b_1 \ldots b_n \in L(\mathcal{A}, \varphi)$, the sentence $\psi_{\mathcal{A},\varphi}$ states the existence of an accepting run $\varrho = q_0 \ldots q_n \in Q^*$ on a word $(b_1, \overline{d}_1) \ldots (b_n, \overline{d}_n) \in (\Sigma \times D)^*$. In particular, $q_0 = q_{\mathrm{I}}$, $q_{i+1} \in \delta(q_i, (b_{i+1}, \overline{d}_{i+1}))$, for $0 \leq i < n$, and $q_n \in F$. We code such a state sequence ϱ by pairwise disjoint sets $Y_1, \ldots, Y_r \subseteq \{0, \ldots, n\}$ such that Y_q contains those positions of w where state q is assumed, i.e. $i \in Y_q$ iff $q_i = q$. Moreover, we keep track of the numbers at the kth position of the vectors $\overline{d}_i$ with the sets $Z_k^0, \ldots, Z_k^K \subseteq \{1, \ldots, n\}$: it holds $i \in Z_k^d$ iff the kth position of $\overline{d}_i$ is d. We require that $0 \in Z_k^0$. Therefore, the kth position of the vector $\overline{d}_1 + \cdots + \overline{d}_n$ is $\sum_{0\leq d\leq K} d|Z_k^d|$. Formally,

$$
\begin{aligned}
&\exists Z_1^0 \ldots \exists Z_1^K \ldots \exists Z_N^0 \ldots \exists Z_N^K \exists Y_1 \ldots \exists Y_r \\
&\exists n \exists n' \Big(\mathrm{succ}(n, n') \wedge \psi_{\max}(n, U_1, \ldots, U_s) \wedge \\
&\quad \textstyle\bigwedge_{1\leq i\leq N} \psi_{\mathrm{part}}(n', Z_i^0, \ldots, Z_i^K) \wedge \psi_{\mathrm{part}}(n', Y_1, \ldots, Y_r) \wedge \\
&\quad \textstyle\bigwedge_{1\leq i\leq N} Z_i^0 0 \wedge Y_{q_{\mathrm{I}}} 0 \wedge \\
&\quad \forall x \forall x' \Big(x < n' \wedge \mathrm{succ}(x, x') \rightarrow \\
&\qquad\qquad\qquad \textstyle\bigvee_{q\in\delta(p,(b,(d_1,\ldots,d_N)))} (Y_p x \wedge U_b x \wedge Y_q x' \wedge \bigwedge_{1\leq j\leq N} Z_j^{d_j} x') \Big) \wedge \\
&\quad \textstyle\bigvee_{q\in F} Y_q n' \wedge \varphi^*[\sum_{0\leq d\leq K} d|Z_1^d|/z_1, \ldots, \sum_{0\leq d\leq K} d|Z_N^d|/z_N] \Big) \,,
\end{aligned}
$$

where φ^* is inductively defined. We only state the base cases:

$$(t < t')^* := t < t'$$

and

$$
\begin{aligned}
(t \equiv t' \mod n)^* := {}& \exists x \exists y \exists X \exists Y \big(\forall z (Xz \leftrightarrow z < x) \wedge \forall z (Yz \leftrightarrow z < y) \wedge \\
& |X| = t \wedge |Y| = t' \wedge x \equiv y \mod n \big) \,.
\end{aligned}
$$

The relation $x \equiv y \bmod n$, for a fixed $n \geq 2$, can be expressed by a WS1S formula. ⊣

From Theorem 11 we obtain the following undecidability result.

Corollary 12. WS1S$^{\mathrm{card}}$ *is undecidable. Already the fragment consisting of the sentences of the form* $\forall\overline{U}\exists\overline{X}\chi$ *is undecidable, where all bounded variables in* χ *are FO variables.*

3.3 WS1S$^{\mathrm{card}}$ and Parikh Automata

The two theorems — Theorem 11 in §3.2 and Theorem 13 in this subsection — establish the automata-logic connection between Parikh automata and the fragment $\mathsf{F}_\exists$ of WS1S$^{\mathrm{card}}$. Theorem 16 shows decidability of the fragment consisting of the sentences of the form $Q_1 x_1 \dots Q_m x_m \varphi$ with $Q_1, \dots, Q_m \in \{\exists, \forall\}$ and $\varphi \in \mathsf{F}_\exists \cup \mathsf{F}_\forall$.

Theorem 13. *For every formula $\varphi \in \mathsf{F}_\exists$, we can construct a Parikh automaton recognizing $L(\varphi)$.*

To prove this theorem we need the following two lemmas.

Lemma 14. *Let $QX_1 \dots QX_n \psi \in \mathsf{F}_\exists \cup \mathsf{F}_\forall$. The formula ψ can be effectively transformed into a logically equivalent formula of the form $\bigvee_i \bigwedge_j \psi_{ij}$, where each ψ_{ij} is either a WS1S formula or $\psi_{ij} \in \mathsf{C}$.*

Lemma 15. *For every $\psi \in \mathsf{C}$ there is a Parikh automaton recognizing $L(\psi)$.*

Proof (Theorem 13). Let $\varphi(\alpha_1, \dots, \alpha_m) \in \mathsf{F}_\exists$. Because of Lemma 14 we can assume that $\varphi = \exists X_1 \dots \exists X_n (\bigvee_i \bigwedge_j \psi_{ij})$, where ψ_{ij} is either a WS1S formula or $\psi_{ij} \in \mathsf{C}$.

By Fact 9 we can construct an automaton $\mathcal{A}_{ij}$ with $L(\mathcal{A}_{ij}) = L(\psi_{ij})$ if ψ_{ij} is a WS1S formula, and, because of Lemma 15, if $\psi_{ij} \in \mathsf{C}$ we can construct a Parikh automaton $(A_{ij}, \widehat{\psi}_{ij})$ such that $L(\mathcal{A}_{ij}, \widehat{\psi}_{ij}) = L(\psi_{ij})$. Since Parikh automata are closed under union and intersection, we can construct a Parikh automaton $(\mathcal{A}, \widehat{\psi})$ recognizing $L(\bigvee_i \bigwedge_j \psi_{ij})$.

Now, we construct a Parikh automaton $(\mathcal{B}, \widehat{\varphi})$ recognizing $L(\varphi)$. Intuitively, the desired Parikh automaton $(\mathcal{B}, \widehat{\varphi})$ acts as $(\mathcal{A}, \widehat{\psi})$ for $\bigvee_i \bigwedge_j \psi_{ij}$ but $\mathcal{B}$ is allowed to guess the tracks for the variables $X_1, \dots, X_n$. Let $\mathcal{A}'$ be the result of such a projection operation on $\mathcal{A}$. Unfortunately, $\mathcal{A}'$ does not quite do the job. The problem is that the guessed track, encoding the interpretation for a variable X_i, may be longer than a word w encoding the interpretation for the free variables of φ. Therefore, a Parikh automaton has to accept all words that can be extended by a word in $(\{0\}^m)^*$ that are accepted by $(\mathcal{A}, \widehat{\psi})$. Let $(\mathcal{B}, \widehat{\varphi})$ be the Parikh automaton from $(\mathcal{A}, \widehat{\psi})$ that is obtain by a projection from $\{0,1\}^{m+n}$ to $\{0,1\}^m$ and the right quotient with the regular language $(\{0\}^m)^*$. We have $L(\mathcal{B}, \widehat{\varphi}) = L(\varphi)$. ⊣

In the remainder of this section, we will show the following decidability result.

Theorem 16. *The fragment of* WS1S$^{\mathrm{card}}$ *consisting of the sentences of the form $Q_1 x_1 \dots Q_m x_m \varphi$ with $\varphi \in \mathsf{F}_\exists \cup \mathsf{F}_\forall$ is decidable, where $Q_1, \dots, Q_m \in \{\exists, \forall\}$.*

Proof. Case I: $\varphi \in \mathsf{F}_\exists$. Let $\varphi = \exists X_1 \dots \exists X_n \varphi'$ such that all MSO variables occurring in a subformula in C of φ' are bounded by one of the quantifiers for $X_1, \dots, X_n$.

Because of Theorem 13 we can construct a Parikh automaton $(\mathcal{A}, \widehat{\varphi})$ with $L(\varphi') = L(\mathcal{A}, \widehat{\varphi})$. Assume that the dimension of $(\mathcal{A}, \widehat{\varphi})$ is $N \geq 0$ and that $\mathcal{A} = (S, \Gamma, \delta, s_{\mathrm{I}}, F)$, where $\Gamma \subseteq \{0,1\}^{m+n} \times \mathbb{N}^N$.

It holds that $(\mathbb{N}, \mathrm{succ}) \models Q_1 x_1 \dots Q_m x_m Q X_1 \dots Q X_n \varphi'$ is equivalent to

$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there exists a word } w \in L(\mathcal{A}, \widehat{\varphi}) \text{ such that} \\ &\qquad I_w(x_1) = \tilde{x}_1, \dots, I_w(x_m) = \tilde{x}_m\,. \end{aligned} \tag{2}$$

By definition, (2) is equivalent to

$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there exists a word } w \in L(\mathcal{A}) \text{ such that} \\ &\qquad I_{\Psi(w)}(x_1) = \tilde{x}_1, \dots, I_{\Psi(w)}(x_m) = \tilde{x}_m \text{ and } \mathfrak{N} \models \widehat{\varphi}[\Phi(w)/\overline{z}]\,, \end{aligned} \tag{3}$$

where Ψ is the projection from Γ to $\{0,1\}^{m+n}$, and Φ is the Parikh map of Γ.

We extend the alphabet Γ to $\Gamma' := \{(b, \overline{v}, \overline{v'}) \mid (b, \overline{v}) \in \Gamma \text{ and } \overline{v}' \in \{0,1\}^m\}$, i.e., we append the vectors $\overline{v}' \in \{0,1\}^m$ to each symbol $(b, \overline{v}) \in \Gamma$. Let Φ' be the Parikh map of Γ'. In the following, we construct an automaton $\mathcal{A}'$ with alphabet Γ' such that

$$\begin{aligned} L(\mathcal{A}') = \{w \in \Gamma'^* \mid\ & h(w) \in L(\mathcal{A}) \text{ and} \\ & \Phi'(w) = \big(\Phi(h(w)), I_{\Psi(h(w))}(x_1), \dots, I_{\Psi(h(w))}(x_m)\big)\}\,, \end{aligned}$$

where $h : \Gamma'^* \to \Gamma^*$ is the homomorphism defined by $h(b, \overline{v}, \overline{v}') := (b, \overline{v})$.

Let $\mathcal{A}' := (S, \Gamma', \delta', s_{\mathrm{I}}, F)$, where the transition function δ' contains the same transitions as δ except that $\mathcal{A}'$ marks the positions in a word that determine the values of the interpretations for the FO variables $x_1, \dots, x_m$. For each FO variable x_i, let $B_i \subseteq S$ be the set that contains all the states that are reachable *before* reading a symbol that determines the interpretation of the FO variable x_i, i.e. $B_i := \bigcup_{j \in \mathbb{N}} B_i^j$, where $B_i^0 := \{s_{\mathrm{I}}\}$ and for $j > 0$,

$$B_i^{j+1} := \{s \in \delta(B_i^j, (b, \overline{v})) \mid (b, \overline{v}) \in \Gamma \text{ with } \chi_{x_i}(b) = 0\}\,.$$

For $s \in S$, let $\overline{c}(s) \in \{0,1\}^m$ be the characteristic vector of s, i.e. $\overline{c}(s) := (c_1, \dots, c_m)$, where $c_i = 1$ iff $s \in B_i$. Now, $\delta' : S \times \Gamma' \to \mathcal{P}(S)$ is defined by

$$\delta'(s, (b, \overline{v}, \overline{v}')) := \{s' \in \delta(s, (b, \overline{v})) \mid \overline{c}(s') = \overline{v}'\}\,.$$

By the construction of $\mathcal{A}'$, (3) is equivalent to

$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there exists a word } w \in L(\mathcal{A}') \text{ such that} \\ &\qquad \mathfrak{N} \models \widehat{\varphi}[\Phi(h(w))/\overline{z}] \text{ and } \Phi'(w) = (\Phi(h(w)), \tilde{x}_1, \dots, \tilde{x}_m)\,. \end{aligned} \tag{4}$$

From Lemma 5, it follows that $\Phi'(L(\mathcal{A}'))$ is the union of the images of linear, constructible polynomials $q_1, \dots, q_k$. We conclude that (4) is equivalent to

$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there exist } y_1, \dots, y_r \in \mathbb{N} \text{ and } \overline{v} \in \mathbb{N}^N \\ &\qquad \text{such that } \mathfrak{N} \models \widehat{\varphi}[\overline{v}/\overline{z}] \text{ and} \\ &\qquad q_i(y_1, \dots, y_r) = (\overline{v}, \tilde{x}_1, \dots, \tilde{x}_m), \text{ for some } 1 \leq i \leq k\,. \end{aligned} \tag{5}$$

Note that (5) can be expressed as a sentence in Presburger arithmetic. Since Presburger arithmetic is decidable we are done.

<u>Case II:</u> $\varphi \in \mathsf{F}_\forall$. Follows from Case I by the duality of quantifiers. ⊣

4 Parikh Tree Automata

We generalize the framework presented in §2 to trees, and sketch how the results in §3 for $\mathrm{WS1S^{card}}$ carry over to WS2S extended with cardinality constraints.

A (full binary) Γ-labeled **tree** is a function t, where the range of t is the alphabet Γ and the domain of t, $\mathrm{dom}(t)$, is a finite, nonempty, prefix-closed subset of $\{0,1\}^*$, where $u0 \in \mathrm{dom}(t)$ iff $u1 \in \mathrm{dom}(t)$, for all $u \in \mathrm{dom}(t)$. The **subtree** of t with root $u \in \mathrm{dom}(t)$ is denoted by t_u, where $\mathrm{dom}(t_u) := \{v \mid uv \in \mathrm{dom}(t)\}$ and $t_u(v) := t(uv)$. The set of Γ-labeled trees is denoted by T_Γ.

Parikh's theorem carries over to regular tree languages, where the commutative image of a Γ-labeled tree t counts, as for words, the numbers of occurrences of the symbols of Γ in t.

Lemma 17. *The commutative image of a regular tree language is semi-linear.*

Proof. For a top-down tree automaton $\mathcal{A} = (Q, \Gamma, \delta, q_\mathrm{I}, F)$, we construct a context-free grammar G that generates the words obtained by traversing the trees recognized by $\mathcal{A}$ in infix order. The set of nonterminals of G is Q, the start symbol of G is q_I, and we have a derivation rule $p \to qbq'$ for $(q,q') \in \delta(p,b)$. And moreover, $q \to b$ whenever $(F \times F) \cap \delta(q,b) \neq \emptyset$. The claim follows now from Parikh's theorem. ⊣

In the following, let Γ be an alphabet $\Sigma \times D$ where $D \subseteq \mathbb{N}^n$, for some $n \geq 0$. The **projection** (for trees) Ψ maps a Γ-labeled tree t to the Σ-labeled tree t', where $\mathrm{dom}(t') := \mathrm{dom}(t)$ and $t'(u) := b$, for $t(u) = (b, \overline{d})$. The **Parikh map** (for trees) is the mapping $\Phi : T_\Gamma \to \mathbb{N}^n$ recursively defined by $\Phi(t) := \overline{d}$ if $\mathrm{dom}(t) = \{\lambda\}$, and otherwise $\Phi(t) := \overline{d} + \Phi(t_0) + \Phi(t_1)$, where $t(\lambda) = (b, \overline{d})$. The **restriction** $L\restriction_\varphi$ of $L \subseteq T_\Gamma$ with respect to the Presburger arithmetic formula $\varphi(z_1, \ldots, z_n)$ is defined as in Definition 2.

Parikh tree automata are defined analogously to Parikh automata. In the Definition 3 we replace the word automaton by a tree automaton. The closure properties of Parikh automata stated in Property 4 carry over to Parikh tree automata. The decidability of the emptiness problem of Parikh tree automata can be proved analogously to Property 6.

As for WS1S in §3 we can define an extension of WS2S with cardinality constraints which we call $\mathrm{WS2S^{card}}$. Instead of the atomic formula $\mathrm{succ}(x,y)$ in WS1S we have in WS2S the atomic formulas $\mathrm{succ}_0(x,y)$ and $\mathrm{succ}_1(x,y)$ for the left and right successor of the infinite binary tree $\mathbb{T} := \{0,1\}^*$. An interpretation maps now FO variables to elements in $\mathbb{T}$, and MSO variables are mapped to finite subsets of $\mathbb{T}$. For example, the root λ of $\mathbb{T}$ can be defined by $\neg\exists y(\mathrm{succ}_0(y,x) \vee \mathrm{succ}_1(y,x))$. $\mathrm{WS2S^{card}}$ has in addition to the atomic formulas of WS2S the atomic formulas of the form $c+|X_1|+\cdots+|X_r| < d+|Y_1|+\cdots+|Y_s|$, where $c, d \in \mathbb{N}$. Their truth values are defined analogously as for $\mathrm{WS1S^{card}}$.

Analogously to the description of Parikh automata in $\mathrm{WS1S^{card}}$, we can describe Parikh tree automata in $\mathrm{WS2S^{card}}$. $\mathrm{WS2S^{card}}$ is undecidable since it includes $\mathrm{WS1S^{card}}$. The elements of the set $\{0\}^*$ are interpreted as natural numbers, the left successor relation is interpreted as the successor relation on the natural numbers, and FO quantification in $\mathrm{WS2S^{card}}$ is relativized by the formula

$$\psi_{\mathrm{FO}}(x) := \forall Z\big(Zx \wedge \forall z \forall z'(\mathrm{succ}_0(z,z') \wedge Zz' \to Zz) \to Z\lambda\big)$$

and MSO quantification is relativized by the formula

$$\psi_{\mathrm{MSO}}(X) := \forall y(Xy \to \psi_{\mathrm{FO}}(y)) .$$

We write $Qx.(\psi(x))\,\varphi$ for $\forall x(\psi(x) \to \varphi)$ if $Q = \forall$, and $\exists x(\psi(x) \wedge \varphi)$ if $Q = \exists$. As for $\mathrm{WS1S^{card}}$, we can identify a decidable fragment of $\mathrm{WS2S^{card}}$. The proof of the Theorem 16 can be adopted for the following theorem.

Theorem 18. *The fragment of* $\mathrm{WS2S^{card}}$ *consisting of the sentences of the form*

$$Q_1 x_1.(\psi_{\mathrm{FO}}(x_1)) \dots Q_m x_m.(\psi_{\mathrm{FO}}(x_m))\, QX_1 \dots QX_n \varphi ,$$

where $Q_1, \dots, Q_m, Q \in \{\exists, \forall\}$ and all MSO variables occurring in a subformula in C of φ are bounded by one of the quantifiers for $X_1, \dots, X_n$, is decidable.

5 Conclusion

We have extended WS1S and WS2S with cardinality constraints, proved the undecidability of these extensions, and identified decidable fragments of these logics. These results are obtained by representing the set of models of a formula as the language recognized by a Parikh word (tree) automaton. Parikh automata extend classical automata in that their acceptance conditions include arithmetic constraints based on Parikh's commutative image. Elsewhere we demonstrated a useful application of our decidability results [17].

Several other extensions of word automata with counters are found in the literature. Most of them, e.g. [8, 13, 15], restrict the applicability of transitions in a run by additional guards on the values of the counters. Instead, a Parikh automaton constrains the language of an automaton over the extended alphabet with a Presburger arithmetic formula. It turns out that Parikh word automata are as expressive as Ibarra's *reversal-bounded multicounter machines* [15].

Variations of Parikh automata can be obtained by exchanging either the underlying device or the underlying arithmetic logic. For example, Parikh pushdown automata are simply obtained by considering pushdown automata instead of word automata. Parikh pushdown automata are powerful enough to express extensions of reversal-bounded multicounter machines such as one-way multicounter machines, where one counter is not bounded by a number restricting the alternation of increase and decrease of its count [15, 16]. The main concepts and techniques employed in this paper can also be used to extend the logic-automaton connection to the logics S1S and S2S with cardinality constraints by defining Parikh automata on infinite words and trees. Yet another variation is to consider Parikh maps with vectors of real numbers instead of natural numbers, and Presburger arithmetic is replaced with the first-order logic over the linear arithmetic over the reals and integers [3]. We believe that this variation of Parikh automata has applications for modeling dense real-time systems. We plan to investigate all these variations on Parikh automata in the future. Further work will also include detailed complexity analyses, theoretically and practically, on Parikh automata and on the decision procedure for $\mathrm{WS1S}^{\mathrm{card}}$.

Although, WS1S is only non-elementary decidable [21, 28], we are encouraged by the results obtained by Mona's highly specialized and tuned automata-based decision procedure for WS1S [18]. In many case studies it is shown that Mona behaves well in practice. Our decision procedure for $\mathrm{WS1S}^{\mathrm{card}}$ is even more complex than Mona's decision procedure for WS1S in the sense that we do not only have to construct corresponding automata for the subformulas of a given $\mathrm{WS1S}^{\mathrm{card}}$ sentence but we also have to decide Presburger arithmetic. The automata constructions needed for an implementation of our decision procedure are mainly the constructions used in Mona. Presburger arithmetic can be encoded in WS1S. Again, practice shows that the running times based on such an encoding and using Mona are reasonable [27]. A problem may become the construction of the Presburger arithmetic formulas since we have to determine the polynomials of the commutative image of a word automaton.

Petri net reachability is expressible in $\mathrm{WS1S}^{\mathrm{card}}$ [17], but the encoding requires a top-level quantification of the form $\exists x \exists \overline{X} \forall y \exists \overline{Y}$, which falls slightly outside the scope of our decision procedure. Decidability of Petri net reachability, however, suggests that such a fragment of $\mathrm{WS1S}^{\mathrm{card}}$ may still be decidable.

References

1. D. Basin and S. Friedrich, *Combining WS1S and HOL*, in FroCos'98, Applied Logic Series, 2000, pp. 39–56.
2. D. Basin and N. Klarlund, *Automata based symbolic reasoning in hardware verification*, Formal Methods in Systems Design, 13 (1998), pp. 255–288.
3. B. Boigelot, S. Jodogne, and P. Wolper, *On the use of weak automata for deciding linear arithmetic with integer and real variables*, in IJCAR'01, vol. 2083 of LNCS, 2001, pp. 611–625.
4. A. Bouajjani and P. Habermehl, *Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations*, Theoretical Computer Science, 221 (1999), pp. 211–250.
5. V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire, *Logic and p-recognizable sets of integers*, Bull. Belg. Math. Soc., 1 (1994), pp. 191–238.
6. J. Büchi, *Weak second-order arithmetic and finite automata*, Zeitschrift der mathematischen Logik und Grundlagen der Mathematik, 6 (1960), pp. 66–92.
7. ———, *On a decision method in restricted second-order arithmetic*, in Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science, Stanford, 1962, Stanford Univ. Press, pp. 1–11.
8. H. Comon and Y. Jurski, *Multiple counters automata, safety analysis and Presburger arithmetic*, in CAV'98, vol. 1427 of LNCS, 1998, pp. 268–279.

9. J. Doner, *Tree acceptors and some of their applications*, Journal of Computer and System Sciences, 4 (1970), pp. 406–451.
10. J. Elgaard, N. Klarlund, and A. Møller, *Mona 1.x: New techniques for WS1S and WS2S*, in CAV'98, vol. 1427 of LNCS, 1998, pp. 516–520.
11. J. Elgaard, A. Møller, and M. Schwartzbach, *Compile-time debugging of C programs working on trees*, in ESOP'00, vol. 1782 of LNCS, 2000, pp. 119–134.
12. C. Elgot, *Decision problems of finite automata design and related arithmetics*, Transactions of the AMS, 98 (1961), pp. 21–51.
13. A. Finkel and G. Sutre, *Decidability of rechability problems for classes of two counter automata*, in STACS'00, vol. 1770 of LNCS, 2000, pp. 346–357.
14. J. Henriksen, J. Jensen, M. Jorgensen, N. Klarlund, B. Paige, T. Rauhe, and A. Sandholm, *Mona: Monadic second-order logic in practice*, in TACAS'95, vol. 1019 of LNCS, 1995, pp. 89–110.
15. O. Ibarra, *Reversal-bounded multicounter machines and their decision problems*, Journal of the ACM, 25 (1978), pp. 116–133.
16. O. Ibarra, J. Su, Z. Dang, T. Bultan, and R. Kemmerer, *Counter machines: Decidable properties and applications to verification problems*, in MFCS'00, vol. 1893 of LNCS, 2000, pp. 426–435.
17. F. Klaedtke and H. Ruess, *WS1S with cardinality constraints*, Tech. Rep. SRI-CSL-05-01, SRI International, 333 Ravenswood, Menlo Park, CA, 94025, USA, November 2001.
18. N. Klarlund, A. Møller, and M. Schwartzbach, *MONA implementation secrets*, in CIAA'00, vol. 2088 of LNCS, 200, pp. 182–194.
19. N. Klarlund, M. Nielsen, and K. Sunesen, *Automated logical verification based on trace abstraction*, in PODC'96, 1996, pp. 101–110.
20. L. Lamport, R. Shostak, and M. Pease, *The Byzantine Generals problem*, ACM Transactions on Programming Languages and Systems, 4 (1982), pp. 382–401.
21. A. Meyer, *Weak monadic second-order theory of successor is not elementary-recursive*, in Logic Colloquium, vol. 453 of Lecture Notes in Mathematics, 1975, pp. 132–154.
22. S. Owre and H. Ruess, *Integrating WS1S with PVS*, in CAV'00, vol. 1855 of LNCS, 2000, pp. 548–551.
23. R. Parikh, *On context-free languages*, Journal of the ACM, 13 (1966), pp. 570–581.
24. M. Rabin, *Decidability of second-order theories and automata on infinite trees*, Transactions of the AMS, 141 (1969), pp. 1–35.
25. J. Rushby, *Systematic formal verification for fault-tolerant time-triggered algorithms*, IEEE Transactions on Software Engineering, 2 (1999), pp. 651–660.
26. M. Schützenberger, *On the definition of a family of automata*, Information and Control, 4 (1961), pp. 245–270.
27. T. Shiple, J. Kukula, and R. Ranjan, *A comparison of Presburger engines for EFSM reachability*, in CAV'98, vol. 1427 of LNCS, 1998, pp. 280–292.
28. L. Stockmeyer, *The Complexity of Decision Problems in Automata Theory and Logic*, PhD thesis, Dept. of Electrical Engineering, MIT, Boston, Mass., 1974.
29. J. Thatcher and J. Wright, *Generalized finite automata theory with an application to a decision problem in second-order logic*, Mathematical System Theory, 2 (1968), pp. 57–81.
30. W. Thomas, *Languages, automata, and logic*, in Handbook of Formal Languages, A. Salomaa and G. Rozenberg, eds., vol. 3, Springer-Verlag, 1997, ch. 7, pp. 389–455.
31. B. Trakhtenbrot, *Finite automata and the logic of one-place predicates*, AMS, Transl., II. Ser., 59 (1966), pp. 23–55.

A Additional Details to §2

We recall the definition of finite word automata. A **(finite word) automaton** $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, q_{\mathrm{I}}, F)$, where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ is the transition function, q_{I} is the initial state, and $F \subseteq Q$ is the set of final states. A **run** ϱ of $\mathcal{A}$ on $b_1 \dots b_n \in \Sigma^*$ is a word $q_1 \dots q_{n+1} \in Q^*$ with $q_1 = q_{\mathrm{I}}$ and $q_{i+1} \in \delta(q_i, b_i)$, for all $1 \leq i < n$. ϱ is **accepting** if $q_{n+1} \in F$. A word w is **recognized** by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. $L(\mathcal{A})$ denotes the set of words that are recognized by $\mathcal{A}$.

A.1 Closure Properties

In this subsection, we give the proofs of the closure properties that we have stated in §2. Moreover, we show that Parikh automata are not closed under complement, and that deterministic Parikh automata are not closed under homomorphisms.

Theorem 19. *Parikh automata and deterministic Parikh automata are effectively closed under intersection, union, and inverse homomorphisms.*

Proof. Intersection: Let $(\mathcal{A}_1, \varphi_1)$ and $(\mathcal{A}_1, \varphi_1)$ be Parikh automata of dimension $N_1 \geq 0$ and $N_2 \geq 0$, respectively, where $\mathcal{A}_i = (Q_i, \Gamma_i, \delta_i, q_{\mathrm{I}i}, F_i)$ with $\Gamma_i = \Sigma \times D^{N_i}$, for $i \in \{0, 1\}$. We give a construction for a Parikh automaton $(\mathcal{B}, \psi)$ of dimension $N_1 + N_2$ recognizing $L(\mathcal{A}_1, \varphi_1) \cap L(\mathcal{A}_2, \varphi_2)$.

Let $\mathcal{B} := (Q_1 \times Q_2, \Sigma \times D_1 \times D_2, \delta, (q_{\mathrm{I}1}, q_{\mathrm{I}2}), F_1 \times F_2)$, where the transition function of $\mathcal{B}$ is defined by

$$\delta\big((q_1, q_2), (b, \overline{v}_1, \overline{v}_2)\big) := \delta_1\big(q_1, (b, \overline{v}_1)\big) \times \delta_2\big(q_2, (b, \overline{v}_2)\big)$$

for $q_i \in Q_i$ and $(b, \overline{v}_i) \in \Gamma_i$, for $i \in \{1, 2\}$. Note that if $(\mathcal{A}_1, \varphi_1)$ and $(\mathcal{A}_2, \varphi_2)$ are deterministic then $(\mathcal{B}, \psi)$ is also deterministic. Moreover, let

$$\psi(z_1, \dots, z_{N_1+N_2}) := \varphi_1(z_1, \dots, z_{N_1}) \wedge \varphi_2[z_{N_1+1}/z_1, \dots, z_{N_1+N_2}/z_{N_2}]\,.$$

It is straightforward to verify that $(\mathcal{B}, \psi)$ recognizes $L(\mathcal{A}_1, \varphi_1) \cap L(\mathcal{A}_2, \varphi_2)$.

Union: Let $(\mathcal{A}_1, \varphi_1)$ and $(\mathcal{A}_2, \varphi_2)$ be as above. Without loss of generality, we assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are complete, i.e., $\delta_i(q, b, \overline{v}) \neq \emptyset$, for every $q \in Q_i$ and $(b, \overline{v}) \in \Gamma_i$. For simplicity, we assume that $\lambda \notin L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$. We construct a Parikh automaton $(\mathcal{B}, \varphi)$ of dimension $N := N_1 + N_2 + 2$ that recognizes $L(\mathcal{A}_1, \varphi_1) \cup L(\mathcal{A}_2, \varphi_2)$.

Let $\mathcal{B} := (Q_1 \times Q_2, \Gamma, \delta, (q_{\mathrm{I}1}, q_{\mathrm{I}2}), (F_1 \times Q_2) \cup (Q_1 \times F_2))$, where $\Gamma := \Sigma \times D_1 \times D_2 \times \{0, 1, 2, 3\}^2$ and the transition function is defined as follows. Intuitively, $\mathcal{B}$ can make a transition from (p_1, p_2) to (q_1, q_2) via $(b, \overline{v_1}, \overline{v}_2, c_1, c_2)$ if $\mathcal{A}_i$ can make a transition from p_i to q_i via $(b, \overline{v}_i)$, and c_1, c_2 keep track whether q_i is a finial state of $\mathcal{A}_i$, for $i \in \{0, 1\}$. Formally,

$$\begin{aligned}
&\delta\big((p_1, p_2), (b, \overline{v}_1, \overline{v}_2, c_1, c_2)\big) := \\
&\qquad \big\{(q_1, q_2) \mid q_i \in \delta_i\big(q_i, (b, \overline{v}_i)\big) \text{ for every } i \in \{0, 1\}, \text{ and} \\
&\qquad\qquad (c_1, c_2) = \mathrm{status}_1(p_1, q_1) + \mathrm{status}_2(p_2, q_2)\big\}\,,
\end{aligned}$$

where $\mathrm{status}_i : Q_i \times Q_i \to \mathbb{N} \times \mathbb{N}$ is the function given by

$$\mathrm{status}_i(p,q) := \begin{cases} (2^{i-1},0) & \text{if } p \notin F_i \text{ and } q \in F_i, \\ (0,2^{i-1}) & \text{if } p \in F_i \text{ and } q \notin F_i, \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \{0,1\}$. Moreover, let

$$\begin{aligned} &\psi(z_1,\dots,z_N) := \\ &\qquad \big((z_{N-1} = z_N + 1 \vee z_{N-1} = z_N + 3) \wedge \varphi_1(z_1,\dots,z_{N_1})\big) \vee \\ &\qquad \big((z_{N-1} = z_N + 2 \vee z_{N-1} = z_N + 3) \wedge \varphi_2[z_{N_1+1}/z_1,\dots,z_{N_1+N_2}/z_{N_2}]\big)\,. \end{aligned}$$

It is straightforward to show that $L(\mathcal{B},\varphi) = L(\mathcal{A}_1,\varphi_1) \cup L(\mathcal{A}_2,\varphi_2)$. Note that if $(\mathcal{A}_1,\varphi_1)$ and $(\mathcal{A}_2,\varphi_2)$ are deterministic then $(\mathcal{B},\psi)$ is also deterministic.

Inverse homomorphisms: Let $h : \Sigma'^* \to \Sigma^*$ be a homomorphisms, and let $(\mathcal{A},\varphi)$ be a Parikh automaton of dimension $N \geq 0$ with $\mathcal{A} = (Q,\Gamma,\delta,q_\mathrm{I},F)$ and $\Gamma \subseteq \Sigma \times \mathbb{N}^N$.

We only give an informal description of the automaton $\mathcal{B}$ of the Parikh automaton $(\mathcal{B},\varphi)$ of dimension N that recognizes $h^{-1}(L(\mathcal{A},\varphi))$. $\mathcal{B}$ has the same set of states, the same initial state, and the same set of final states as $\mathcal{A}$. Assume that $h(b') = b_1 \dots b_n$. $\mathcal{B}$ can make a transition from $p \in Q$ to $q \in Q$ via $(b',\overline{v})$ iff $\mathcal{A}$ can reach the state q from state p via a word $(b_1,\overline{v_1}) \dots (b_n,\overline{v}_n)$, for some $\overline{v}_1,\dots,\overline{v}_n \in \mathbb{N}^N$ with $\overline{v} = \overline{v}_1 + \cdots + \overline{v}_n$. ⊣

Theorem 20. *Parikh automata are effectively closed under homomorphisms.*

Proof. Let $h : \Sigma^* \to \Sigma'^*$ be a homomorphism, and let $(\mathcal{A},\varphi)$ be a Parikh automaton of dimension $N \geq 0$, where $\Gamma \subseteq \Sigma \times \mathbb{N}^N$ is the alphabet of $\mathcal{A}$. Let $\Gamma' := (\Sigma' \cup \{\varepsilon\}) \times (\{\overline{0}\} \cup \{\overline{v} \,|\, (b,\overline{v}) \in \Gamma \text{ for some } b \in \Sigma\})$ where ε is a fresh symbol. We extend h to the homomorphism $h' : \Gamma^* \to \Gamma'^*$ defined by

$$h'(b,\overline{v}) := \begin{cases} (b_1,\overline{v})(b_2,\overline{0}) \dots (b_n,\overline{0}) & \text{if } h(b) = b_1 \dots b_n, \\ (\varepsilon,\overline{v}) & \text{if } h(b) = \lambda, \end{cases}$$

We can apply the standard construction to $\mathcal{A}$ and h' to construct an automaton $\mathcal{A}' = (Q,(\Sigma \cup \{\varepsilon\}) \times D, q_\mathrm{I},\delta,F)$ with $L(\mathcal{A}') = h'(L(\mathcal{A}))$. Now, we have to eliminate the introduced letter ε. In the following, we give the construction of the Parikh automaton $(\mathcal{B},\psi)$ of dimension $N + |Q| - 1$ recognizing $h(L(\mathcal{A},\varphi))$. It is based on the standard construction for eliminating ϵ-transitions in a nondeterministic finite word automaton. Without loss of generality, we assume that $q_\mathrm{I} \notin \delta(Q,(\Sigma \cup \{\varepsilon\}) \times D)$ and $Q = \{0,\dots,r\}$ with $q_\mathrm{I} = 0$. Let $\Phi : ((\Sigma \cup \{\varepsilon\}) \times D)^* \to \mathbb{N}^N$ be the Parikh map.

Let $D' := \{\Phi(w) \,|\, w \in (\{\varepsilon\} \times D)^*(\Sigma \times D)(\{\varepsilon\} \times D)^* \text{ and } |w| \leq 2|Q| + 1\}$. Note that D' is finite. Let $\mathcal{B} := (Q,\Gamma,q_\mathrm{I},\eta,F \cup \{q_\mathrm{I}\})$ with $\Gamma := \Sigma \times D' \times \{0,1\}^{|Q|}$, and $\eta : Q \times \Gamma \to \mathcal{P}(Q)$ is defined as follow for $p,q \in Q$, $b \in \Sigma$, $\overline{d} \in D'$, and $\overline{v} \in \{0,1\}^r$: $q \in \eta(p,(b,\overline{d},\overline{v}))$ iff there are states $p_0,\dots,p_m,q_0,\dots,q_n \in Q$ with $m,n \leq |Q|$, and $\overline{c}_1,\dots,\overline{c}_m,\overline{c},\overline{d}_1,\dots,\overline{d}_n \in D$ such that

- $p = p_0$ and $q = q_n$,
- $p_k \in \delta(p_{k-1}, (\varepsilon, \overline{c}_k))$, for all $1 \leq k \leq m$,
- $q_0 \in \delta(p_m, (b, \overline{c}))$,
- $q_k \in \delta(q_{k-1}, (\varepsilon, \overline{d}_k))$, for all $1 \leq k \leq n$,
- $\overline{d} = \overline{c}_1 + \cdots + \overline{c}_m + \overline{c} + \overline{d}_1 + \cdots + \overline{d}_n$, and
- for a state $\widetilde{q} \in Q \setminus \{q_{\mathrm{I}}\}$, the $\widetilde{q}$th coordinate of $\overline{v}$ is 1 iff $\widetilde{q} = p_i$, for some $1 \leq i \leq m$, or $\widetilde{q} = q_j$, for some $0 \leq j \leq n$.

Intuitively, the additional dimension for $\widetilde{q} \in Q \setminus \{q_{\mathrm{I}}\}$ stores whether $\widetilde{q}$ has been visited in a run. We do not have to store this information for q_{I} since q_{I} is visited in every run.

For the definition of the Presburger arithmetic formula φ, we need the following definitions for $q \in Q$,

$$C_q := \{\overline{0}\} \cup \{\Phi(w) \mid w \in (\{\varepsilon\} \times D)^* \text{ and } q \in \delta(q, w)\}$$

and

$$C'_q := \{\Phi(w) \mid w \in (\{\varepsilon\} \times D)^* \text{ and } q \in \delta(q_{\mathrm{I}}, w)\}\,.$$

By Lemma 5, we know that C_q and C'_q are semi-linear. Therefore, there are Presburger arithmetic formulas $\psi_q(\overline{x})$ and $\psi'_q(\overline{x})$ such that $C_q = \{\overline{d} \in \mathbb{N}^N \mid \mathfrak{N} \models \psi_q[\overline{d}/\overline{x}]\}$ and $C'_q = \{\overline{d} \in \mathbb{N}^N \mid \mathfrak{N} \models \psi'_q[\overline{d}/\overline{x}]\}$. Let $\psi(\overline{z}, z_{N+1}, \ldots, z_{N+r}) := \chi_1 \vee \chi_2$ with

$$\chi_1 := \bigwedge_{q \in Q \setminus \{q_{\mathrm{I}}\}} z_{N+q} = 0 \wedge \big(\bigvee_{q \in Q} \exists \overline{x} \psi'_q(\overline{x}) \big)$$

and

$$\begin{aligned} \chi_2 := \big(& \bigvee_{q \in Q \setminus \{q_{\mathrm{I}}\}} z_{N+q} > 0 \big) \wedge \\ & \exists \overline{x}^0 \ldots \exists \overline{x}^r \Big(\bigwedge_{q \in Q \setminus \{q_{\mathrm{I}}\}} \big(\overline{x}^q > \overline{0} \rightarrow z_{N+q} > 0 \big) \wedge \bigwedge_{q \in Q} \psi_q(\overline{x}^q) \wedge \\ & \qquad \varphi[\overline{z} + \overline{x}^0 + \cdots + \overline{x}^r / \overline{z}] \Big)\,. \end{aligned}$$

The formula χ_1 takes care of the empty word because $\bigwedge_{q \in Q \setminus \{q_{\mathrm{I}}\}} z_{N+q} = 0$ is only true for the word λ. And χ_2 takes care of all nonempty words. ⊣

Theorem 21. *Parikh automata are effectively closed under concatenations.*

Proof. Let $(\mathcal{A}, \varphi)$ and let $(\mathcal{B}, \psi)$ be Parikh automata of the dimensions $M \geq 0$ and $N \geq 0$, respectively. It is straightforward to adopt the standard word automata construction to construct a Parikh automaton of dimension $M + N$ that recognizes $L(\mathcal{A}, \varphi) L(\mathcal{B}, \psi)$. ⊣

The right quotient of a language $L \subseteq \Sigma^*$ with $L' \subseteq \Sigma^*$ is defined as

$$L/L' := \{w \in \Sigma^* \mid \text{it exists a } w' \in L' \text{ such that } ww' \in L\}\,.$$

Theorem 22. *Parikh automata are effectively closed under left and right quotients.*

Proof. It suffices to prove that if a set $\mathcal{L}$ of languages is closed under concatenations and homomorphisms then $\mathcal{L}$ is closed under right quotients. Let $L, L' \subseteq \Sigma^*$ be languages in $\mathcal{L}$. We first tag the elements in L' by the homomorphism $h : \Sigma^* \to \Gamma^*$ with $\Gamma := \{\dot{b} \mid b \in \Sigma\}$ that is given by $h(b) := \dot{b}$, for $b \in \Sigma$. Further, let $g : (\Sigma \cup \Gamma)^* \to \Sigma^*$ be the homomorphism given by $g(b) := b$ and $g(\dot{b}) := \lambda$, for $b \in \Sigma$. It holds that $L/L' = Lg(h(L'))$. Since $\mathcal{L}$ is closed under concatenations and homomorphisms, we have that $L/L' \in \mathcal{L}$.

The proof can be easily adopted to show closure under left quotients. ⊣

Theorem 23. *Deterministic Parikh automata are effectively closed under complement.*

Proof. Let $(\mathcal{A}, \varphi)$ be a Parikh automaton of dimension $N \geq 0$, where $\mathcal{A} = (Q, \Gamma, \delta, q_\mathrm{I}, F)$ and $\Gamma \subseteq \Sigma \times \mathbb{N}^N$. To simplify matters, we assume that $\lambda \notin L(\mathcal{A})$. We construct a Parikh automaton $(\mathcal{B}, \psi)$ of dimension $N + 2$ that recognizes $\Sigma^* \setminus L(\mathcal{A}, \varphi)$.

Let $\psi := z_{N+1} = z_{N+2} \vee (z_{N+1} = z_{N+2} + 1 \wedge \neg\varphi(z_1, \ldots, z_N))$, and let $\mathcal{B} := (Q, \Gamma \times \{0,1\}^2, \delta', q_\mathrm{I}, Q \setminus F)$, where the transition function is defined by

$$\delta'\big(q, (b, \overline{v}, c_1, c_2)\big) := \begin{cases} \delta\big((q, (b, \overline{v})\big) & \text{if } (c_1, c_2) = \mathrm{status}_1\big(q, \delta\big(p, (b, \overline{v})\big)\big), \\ \emptyset & \text{otherwise,} \end{cases}$$

where status_1 is the function defined in the proof of Theorem 19. ⊣

In the following, we show that Parikh automata are not closed under complement and deterministic Parikh automata are not closed under homomorphisms. We need the following lemma.

Lemma 24. *There is no Parikh automaton that recognizes the language*

$$L := \{ww \mid w \in \{a, b\}^*\}\,.$$

Proof. For the sake of absurdity, assume that the Parikh automaton $(\mathcal{A}, \varphi)$ of dimension $N \geq 0$ recognizes L. Let $\mathcal{A} = (Q, \Gamma, \delta, q_\mathrm{I}, F)$ where $\Gamma \subseteq \{a, b\} \times \mathbb{N}^N$, and let Φ be the Parikh map of Γ, and let Ψ be the projection from Γ to $\{a, b\}$.

In the following, we call a pair $(q, \overline{d})$ a **configuration** of $w \in \{a, b\}^*$ if there is a word $w' \in \Gamma^*$ with $\Psi(w') = w$, $\Phi(w') = \overline{d}$, and $q \in Q$ is a state that is reached by $\mathcal{A}$ after reading w'.

For a word $w \in \{a, b\}^*$, let $c(w)$ denote a configuration $(q, \overline{d})$ such that there is a word $w' \in \Gamma^*$ with (i) $\Psi(w') = w$, (ii) there is an accepting run of $\mathcal{A}$ on w' from q, and (iii) $\mathfrak{N} \models \varphi[\overline{d} + \Phi(w')/\overline{z}]$. Note that such a configuration always exists because $ww \in L(\mathcal{A}, \varphi)$. For words $w, w' \in \{a, b\}$ of same length, it must be the case that $c(w) \neq c(w')$, since otherwise $ww' \in L(\mathcal{A}, \varphi)$. It follows that for every $n \geq 0$, $|\{c(w) \mid w \in \{a, b\}^* \text{ and } |w| = n\}| = 2^n$.

For $n \geq 0$, let C_n be the set of configurations $(q, \Phi(w))$, where w is a word over Γ of length n and $q \in Q$ is a state that is reached after $\mathcal{A}$ has read w. We obtain a contradiction by showing that there is an $n \in \mathbb{N}$ with $|C_n| < 2^n$.

Let r be the maximal number of outgoing edges the transition graph of $\mathcal{A}$, i.e. $r := \max\{|\delta(q,x)| \mid q \in Q \text{ and } x \in \Gamma\}$. A combinatorial argument shows that for $n \geq 0$, it holds

$$|C_n| \leq |Q| \binom{|Q| + r + n - 1}{n} .$$

We have,

$$|Q| \binom{|Q| + r + n - 1}{n} = |Q| \tfrac{(n+1) \cdot \cdots \cdot (|Q|+r+n-1)}{(|Q|+r)!} \leq (|Q| + r + n - 1)^{|Q|+r} .$$

Since $|Q|$ and r are fixed there is an $n \in \mathbb{N}$ with $(|Q| + r + n - 1)^{|Q|+r} < 2^n$. $\dashv$

Theorem 25. *Parikh automata are not closed under complement.*

Proof. With Lemma 24 it remains to show that there is a Parikh automaton that recognizes the complement of L. Note that $\{a,b\}^* \setminus L = L_1 \cup L_2$ with

$$L_1 := \{w \in \{a,b\}^* \mid |w| \text{ is odd}\}$$

and

$$L_2 := \{x_1 \dots x_n y_1 \dots y_n \in \{a,b\}^* \mid x_i \neq y_i \text{ for some } 1 \leq i \leq n\} .$$

Since Parikh automata are closed under union we are done when we have given Parikh automata for L_1 and L_2, respectively. For L_1 it is easy to construct a Parikh automata. Let $\mathcal{A}$ be the automaton depicted in Figure 2, and let $\varphi := z_1 = z_2 \wedge z_3 = z_4$. It remains to show that $L(\mathcal{A}, \varphi) = L_2$. Let Φ and Ψ denote the Parikh map and the projection, respectively.

($\supseteq$) Assume that $x_1 \dots x_n y_1 \dots y_n \in L_2$ with $x_i \neq y_i$, for some $1 \leq i \leq n$. It is easy to verify that $\mathcal{A}$ recognizes the word xy, where

$$x := (x_1, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}) \dots (x_{i-1}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix})(x_i, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}) \dots (x_n, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}) ,$$

and

$$y := (y_1, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}) \dots (y_{i-1}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix})(y_i, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}) \dots (y_n, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix})$$

Moreover, $\mathfrak{N} \models \varphi[\Phi(xy)/\overline{z}]$.

($\subseteq$) Assume that $\varrho = q_0 \dots q_n$ is an accepting run of $\mathcal{A}$ on $w := x_1 \dots x_n$. The run ϱ can be decomposed into four pieces: There are indices $0 \leq i_1 < i_2 < i_3 \leq n$

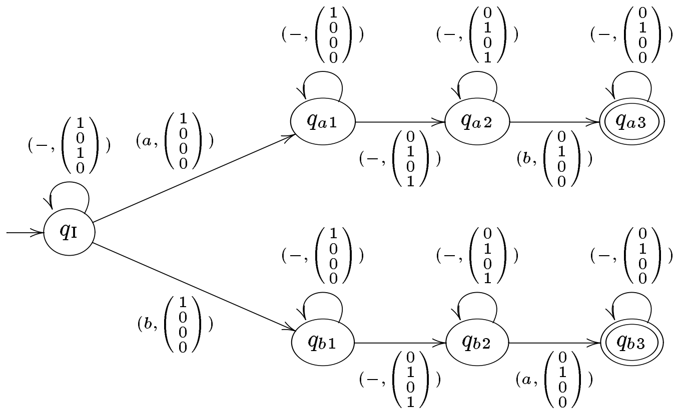

Fig. 2. The automaton for the Parikh automaton that recognizes L_2.

with $q_0 = \dots = q_{i_1} = q_{\mathrm{I}}$, $q_{i_1+1} = \dots = q_{i_2} = q_{x1}$, $q_{i_2+1} = \dots = q_{i_3} = q_{x2}$, and $q_{i_3+1} = \dots = q_n = q_{x3}$, where either $x = a$ or $x = b$. It holds that

$$\begin{aligned}\Phi(w) = \Phi(x_1 \dots x_{i_1}) + \Phi(x_{i_1+1} \dots x_{i_2}) + \Phi(x_{i_2+1} \dots x_{i_3}) + \Phi(x_{i_3+1} \dots x_n) = \\ \begin{pmatrix} i_1 \\ 0 \\ i_1 \\ 0 \end{pmatrix} + \begin{pmatrix} i_2 - i_1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ i_3 - i_2 \\ 0 \\ i_3 - i_2 \end{pmatrix} + \begin{pmatrix} 0 \\ n - i_3 \\ 0 \\ 0 \end{pmatrix} .\end{aligned}$$

Since $\mathfrak{N} \models \varphi[\Phi(w)/\overline{z}]$, we have that $i_2 + i_2 = n$ and $i_1 = i_3 - i_2 = i_3 - n/2$. Hence, $i_3 = i_1 + i_2$. By the construction of $\mathcal{A}$, it follows that $x_{i_1} \neq x_{i_1+i_2}$. Therefore, $\Psi(w) \in L_2$. $\dashv$

Theorem 26. *Deterministic Parikh automata are not closed under homomorphisms.*

Proof. We first observe that L_2 from the proof of Theorem 25 is the homomorphic image of the language

$$\begin{aligned}\widetilde{L}_2 := \{x_1 \dots x_n y_1 \dots y_n \in \{a, b, \widetilde{a}, \widetilde{b}\}^* \mid\ & \text{there is an } 1 \leq i \leq n \text{ such that} \\ & x_1 \dots x_{i-1}, y_1 \dots y_{i-1} \in \{a, b\}^*, \\ & x_i \dots x_n, y_i \dots y_n \in \{\widetilde{a}, \widetilde{b}\}^*, \\ & \text{and } x_i \neq y_i\}\end{aligned}$$

via the homomorphism $h(a) = h(\widetilde{a}) := a$ and $h(b) = h(\widetilde{b}) := b$. It is easy to define a deterministic Parikh automaton that recognizes $\widetilde{L}_2$.

For the sake of absurdity, assume that deterministic Parikh automata are closed under homomorphisms. Therefore, there is a deterministic Parikh automaton that recognizes L_2. With Theorem 19 we conclude that there is a deterministic Parikh automaton that accepts $\{a, b\}^* \setminus L$. Since deterministic Parikh automata are closed under complement there is a deterministic Parikh automaton recognizing L. But this contradicts Lemma 24. $\dashv$

A.2 Universality Problem

We sketch a reduction from the word problem for Turing machines that shows Property 7, i.e., the undecidability of the universality problem for Parikh automata.

Let $\mathcal{M}$ be a Turing machine. Without loss of generality, we assume that $\mathcal{M}$ makes on every input that is accepted by $\mathcal{M}$ an odd number of steps. For any input word w, it holds that $\mathcal{M}$ does not accept w iff all words $c_1\$c_2\$\dots\$c_{m-1}\$c_m$ do not encode a computation of $\mathcal{M}$ on w, where the c_is are configurations of $\mathcal{M}$, \$ is a new symbol, and m is even. Let Σ be the alphabet that contains \$ and all the symbols that can occur in a configuration of $\mathcal{M}$. Let

$$\begin{aligned} L_{\mathcal{M},w} := \{c_1\$\dots\$c_m \in \Sigma^* \mid m \text{ is even, and either} \\ & c_1 \text{ is not an initial configuration of } \mathcal{M} \text{ on } w \\ & \text{or } c_m \text{ is not a final configuration} \\ & \text{or } c_{i+1} \text{ is not a successor configuration of } c_i, \\ & \quad \text{for some } 1 \leq i < m\}\,. \end{aligned}$$

It holds that $\mathcal{M}$ does not accept w iff $L_{\mathcal{M},w} \cup R = \Sigma^*$, where $R \subseteq \Sigma^*$ is the regular language containing the words with an even number of occurrences of \$. It is straightforward to define Parikh automata for the following cases: c_1 is not an initial configuration of $\mathcal{M}$ on w, and c_m is not a final configuration. We can construct a Parikh automaton, analogously as for the language L_2 in Theorem 25, for the case that there is an $1 \leq i < m$ such that c_{i+1} is no successor configuration of c_i, i.e., the Parikh automaton guesses a position in c_i and c_{i+1} where the successor relation is violated. Hence, there is a Parikh automaton recognizing $L_{\mathcal{M},w} \cup R$.

A.3 Reversal-Bounded Multicounter Machines

In this subsection, we recall the definition of reversal bounded multicounter machines [15], and proof that they have the same expressive power as Parikh automata. We start by providing an intuitive description of reversal-bounded multicounter machines.

A *two-way k-counter machine* is a device with a finite state-control, a two-way read-only head which operates on an input tape delimited by endmarkers, and k counters, each capable of storing any natural number. Initially, the device is set to a specified initial state with the read-only head on the left endmarker and all counters are set to 0. An atomic move consists of moving the read-only head $-1, 0, +1$ position to the right, incrementing each counter by $-1, 0, +1$, and changing the control state. The machine prevents the read-only head from falling off the input tape and the counters from storing a negative integer. The input is accepted if the machine eventually reaches a state of a set of final states. Let $m, n \geq 0$. A *(m, n)-reversal-bounded k-counter machine* is a two-way k-counter machine which operates in such a way that in every accepting computation the read-only head reverses direction at most m times and the count in each counter alternately increases and decreases at most n times.

Formally, a **two-way k-counter machine** $\mathcal{M}$ is a 7-tuple $(Q, \Sigma, \delta, q_{\mathrm{I}}, F, \flat, \sharp)$, where

- Q is a finite set of states,
- Σ is an alphabet,
- $\delta : Q \times (\Sigma \cup \{\flat, \sharp\}) \times \{0,1\}^k \to \mathcal{P}(Q \times \{-1, 0, +1\} \times \{-1, 0, +1\}^k)$ is the transition function satisfying the following conditions:
  (i) if $(q, h, \overline{d}) \in \delta(p, \flat, \overline{\chi})$ then $h \in \{0, +1\}$,
  (ii) if $(q, h, \overline{d}) \in \delta(p, \sharp, \overline{\chi})$ then $h \in \{-1, 0\}$,
  (iii) if $(q, h, \overline{d}) \in \delta(p, b, \overline{\chi})$ and $\chi_i = 0$ then $d_i \in \{0, +1\}$, where $1 \leq i \leq k$ and $b \in \Sigma$,
- q_{I} is the initial state,
- $F \subseteq Q$ is the set of final states, and
- $\flat, \sharp \notin \Sigma$ are the left and right endmarkers, respectively.

A **configuration** of $\mathcal{M}$ on the input word $w \in \Sigma^*$ is a tuple $(q, \flat w \sharp, i, \overline{c})$ denoting the fact that $\mathcal{M}$ is in state $q \in Q$ with the read-only head reading the ith letter of $\flat w \sharp$, and $\overline{c} \in \mathbb{N}^k$ are the values of the k counters. We define a relation $\rightsquigarrow$ among configurations as

$$(p, \flat w \sharp, i, \overline{c}) \rightsquigarrow (q, \flat w \sharp, i + h, \overline{c} + \overline{d}) ,$$

where $(q, h, \overline{d}) \in \delta(p, b, \chi(\overline{c}))$, b is the ith letter of $\flat w \sharp$, and $\chi(\overline{c}) \in \{0,1\}^k$ is the vector, where the ith coordinate is 0 iff $c_i = 0$. Note that the conditions (i) and (ii) prevents the read-only head from falling off the input tape, and (iii) ensures that the counters cannot be negative. $\overset{*}{\rightsquigarrow}$ denotes the reflexive, transitive closure of $\rightsquigarrow$. The input word $w \in \Sigma^*$ is **accepted** by $\mathcal{M}$ if $(q_{\mathrm{I}}, \flat w \sharp, 1, \overline{0}) \overset{*}{\rightsquigarrow} (q, \flat w \sharp, i, \overline{c})$, for some $q \in F$, $1 \leq i \leq |\flat w \sharp|$, and $\overline{c} \in \mathbb{N}^k$. We denote with $T(\mathcal{M})$ words that are accepted by $\mathcal{M}$.

Let $m, n \geq 0$. The sequence $(q_1, \flat w \sharp, h_1, \overline{c}_1) \rightsquigarrow \cdots \rightsquigarrow (q_n, \flat w \sharp, h_n, \overline{c}_n)$ is an **(m, n)-computation** of $\mathcal{M}$ on $w \in \Sigma^*$ if

- $q_1 = q_{\mathrm{I}}$, $h_1 = 1$, and $\overline{c}_1 = \overline{0}$,
- the read-only head reverses direction at most m times, i.e., the number of sign alternations in the sequence $(h_2 - h_1) \dots (h_n - h_{n-1}) \in \{-1, 0, +1\}^*$ is at most m,
- the count in each counter alternately increases and decreases at most n times, i.e., the number of sign alternation of the sequence $(c_{i2} - c_{i1}), \dots, (c_{in} - c_{i(n-1)}) \in \{-1, 0, +1\}^*$, for every $1 \leq i \leq k$, is at most n.

An **(m, n)-reversal bounded k-counter machine** $\mathcal{M}$ is a two-way k-counter machine, where every accepting computation is an (m, n)-computation. For instance, a $(0, 1)$-reversal-bounded k-counter machine is equivalent to a one-way k-counter machine which has the property that once a counter is decremented, it can never increase its count again.

Theorem 27. *For every reversal-bounded multicounter machine $\mathcal{M}$ there is a Parikh automaton $(\mathcal{A}, \varphi)$ such that*

$$T(\mathcal{M}) = L(\mathcal{A}, \varphi)\,.$$

Proof (Sketch). In [15] it is shown that a (m, n)-reversal-bounded k-counter machine is equivalent to a $(0, 1)$-reversal-bounded multicounter machine. Therefore, it is sufficient to construct an equivalent Parikh automaton for a $(0, 1)$-reversal-bounded k-counter machine $\mathcal{M} = (Q, \Sigma, \delta, q_{\mathrm{I}}, F, \flat, \sharp)$. We proceed in three steps. In the first step, we construct an equivalent $(0, 1)$-reversal-bounded k-counter machine that only allows $(0, 1)$-computations. In the second step, we construct an equivalent Parikh automaton $(\mathcal{A}, \varphi)$ of dimension $2k + 1$, where the alphabet of $\mathcal{A}$ is infinite. Finally, from $(\mathcal{A}, \varphi)$ we construct an equivalent Parikh automaton $(\mathcal{A}', \varphi')$, where $\mathcal{A}'$ has a finite alphabet.

1. Step: We only sketch the construction of the $(0, 1)$-reversal-bounded k-counter machine that only allows $(0, 1)$-computations.

Since the read-only head of $\mathcal{M}$ does not reverse its direction in all accepting computation, we can assume, without loss of generality, there are no $p, q \in Q$, $\overline{d} \in \{-1, 0, 1\}^k$, $b \in \Sigma$, and $\overline{\chi} \in \{0, 1\}^k$ such that $(q, -1, \overline{d}) \notin \delta(q, b, \overline{\chi})$. Otherwise, we can just delete such a transition.

Assume that there are computations in $\mathcal{M}$ that are not $(0, 1)$-computations. In particular, this means, that there is some $1 \le j \le k$ and some $n \ge 1$ such that there are state $p, p_1, \dots, p_n, q \in Q$ with the following properties:

- $(p_1, h_1, \overline{d}_1) \in \delta(p, b, \overline{\chi})$ with $d_{1j} = -1$,
- $(p_{i+1}, h_{i+1}, \overline{d}_{i+1}) \in \delta(p_i, b_i, \overline{\chi}_i)$ with $d_{(i+1)j} = 0$, for $1 \le i < n$, and
- $(q, h, \overline{d}) \in \delta(p_n, b_n, \overline{\chi}_n)$ with $d_j = 1$,

where $h_1, \dots, h_n, h \in \{0, 1\}$, $\overline{d}_1, \dots, \overline{d}_n, \overline{d} \in \{-1, 0, 1\}^k$, $b, b_1, \dots, b_n \in \Sigma$, and $\overline{\chi}, \overline{\chi}_1, \dots, \overline{\chi}_n \in \{0, 1\}^k$. The upper half of Figure 3 shows the situations for $n = 1$.

Since we know that all accepting computations of $\mathcal{M}$ cannot increment a counter after it has been decremented and a counter must be incremented before it can be decremented, we can transform $\mathcal{M}$ in an equivalent $(0, 1)$-reversal-bounded k-counter machine as depicted in the lower half of Figure 3 for the case where n equals 1. We omit a formal specification of this transformation for $n \ge 1$. By recurring application of the transformation construction, we can assume that all computations of $\mathcal{M}$ are $(0, 1)$-computations.

2. Step: For simplicity, we assume that $\mathcal{M}$ initially moves always its read-only head to the right position and does not change its state and its counters, i.e., $\delta(q_{\mathrm{I}}, \flat, \overline{\chi}) = \{(q_{\mathrm{I}}, 1, \overline{0})\}$, for every $\overline{\chi} \in \{0, 1\}^k$. And we assume that if $\mathcal{M}$ reads the right endmarker then it stops, i.e., $\delta(q, \sharp, \overline{\chi}) = \emptyset$, for every $q \in Q$ and $\overline{\chi} \in \{0, 1\}^k$.

Before we begin with the second construction we have to introduce some notation. For $p, q \in Q$, $b \in \Sigma$, and $\overline{d} \in \mathbb{Z}^k$, we write $p \xrightarrow{b, \overline{d}} q$ if there are $q_1, \dots, q_n \in Q$ and $\overline{c}_0, \dots, \overline{c}_{n+1} \in \mathbb{N}^k$ such that

$$(p, \flat b \sharp, 2, \overline{c}_0) \rightsquigarrow (q_1, \flat b \sharp, 2, \overline{c}_1) \rightsquigarrow \cdots \rightsquigarrow (q_n, \flat b \sharp, 2, \overline{c}_n) \rightsquigarrow (q, \flat b \sharp, 3, \overline{c}_{n+1})\,,$$

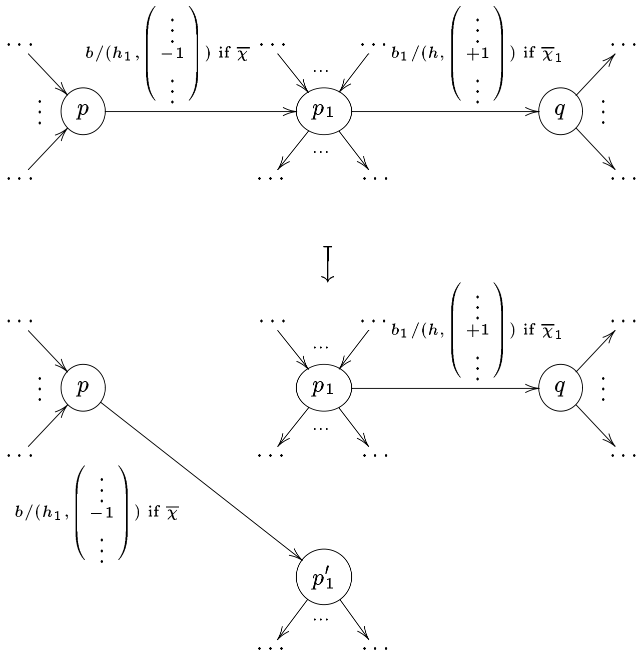

Fig. 3. Transformation in the first construction step.

and $\overline{d} = \overline{c}_{n+1} - \overline{c}_0$. Intuitively, it holds that $p \xrightarrow{b,\overline{d}} q$ if $\mathcal{M}$ can reach the state q from p while moving the read-only head one position to the right and changing the counters with $\overline{d}$.

Let

$$D := \{\overline{d} \in \mathbb{N}^{2k} \mid p \xrightarrow{b,\overline{d}^*} q, \text{ where } p, q \in Q \text{ and } b \in \Sigma\},$$

where $\overline{d}^* := (d_1 - d_2, \dots, d_{2k-1} - d_{2k})$, for $\overline{d} = (d_1, d_2, \dots, d_{2k-1}, d_{2k})$. Note that D may be infinite. For example, if $\mathcal{M}$ possesses a transition that does not change the position of the read-only head but changes at least one counter. D is computable since we can bound n by the number of state of $\mathcal{M}$. Moreover, with a pigeon hole argument one shows that D is of the form $\widehat{D} \cup \bigcup_{1 \leq i \leq m} \{x\overline{d}_i + \overline{d}_i' \mid x \geq 1\}$, for some finite set $\widehat{D} \subseteq \mathbb{N}^{2k}$, some $m \in \mathbb{N}$, and some $\overline{d}_1, \overline{d}_1' \dots, \overline{d}_m, \overline{d}_m' \in \mathbb{N}^{2k}$.

We define a Parikh automaton $(\mathcal{A}, \varphi)$ of dimension $2k+1$, where the alphabet of $\mathcal{A}$ might be infinite. Let $\mathcal{A} := (Q, \Sigma \times D \times \{0,1\}, \eta, q_{\mathrm{I}}, F)$, where the transition function is defined as follows. Let $b \in \Sigma$. For $p \in F$,

$$\eta\big(p, (b, \overline{0}, 0)\big) := \{p\} \qquad \text{and} \qquad \eta\big(p, (b, \overline{d}, 1)\big) := \emptyset$$

for $\overline{d} \neq \overline{0}$. That is, if $\mathcal{A}$ is in a final state of $\mathcal{M}$, then $\mathcal{A}$ stays in this state until it has read the whole input word and the value of the "counters" are not changed.

For $\overline{d} \in \mathbb{N}^{2k}$, $d \in \{0,1\}$, and $p \notin F$,

$$\eta\big(p, (b, \overline{d}, d)\big) := \{q \in Q \mid p \xrightarrow{b,\overline{d}^*} q \text{ and } d = 1 \text{ iff } q \in F\}\,.$$

Intuitively, $\mathcal{A}$ memorizes in its $(2k+1)$st dimension if it has reached a final state. With the dimensions $2i-1$ and $2i$, for $1 \le i \le k$, $\mathcal{A}$ simulates the ith counter of $\mathcal{M}$. The $(2i-1)$st dimension keeps track of the increments of the ith counter, and the $(2i)$th dimension keeps track of the decrements of the ith counter. The Presburger arithmetic formula φ is defined as

$$\varphi(z_1, \ldots, z_{2k+1}) := z_{2k+1} = 1 \wedge \bigwedge_{1 \le i \le k} z_{2i-1} - z_{2i} \ge 0\,,$$

i.e., $\mathcal{A}$ must be in a final state, and the counts of the k counters must be positive. Note that every computation in $\mathcal{M}$ is a $(0,1)$-computation.

3. Step: Now, we construct from $(\mathcal{A}, \varphi)$ an equivalent Parikh automaton $(\mathcal{A}', \varphi')$, where $\mathcal{A}$ has only a finite alphabet. Recall that the alphabet of $\mathcal{A}$ is $\Sigma \times D \times \{0,1\}$ with

$$D = \widehat{D} \cup \bigcup_{1 \le i \le m} \{x\overline{d}_i + \overline{d}_i' \mid x \ge 1\}\,,$$

where $\widehat{D}$ is a finite subset of $\mathbb{N}^{2k}$ and $\overline{d}_1, \overline{d}_1' \ldots, \overline{d}_m, \overline{d}_m' \in \mathbb{N}^{2k}$.

The dimension of $(\mathcal{A}', \varphi')$ is $2k + 1 + m$. The automaton $\mathcal{A}'$ is defined as $\mathcal{A}' := (Q, \Sigma \times D' \times \{0,1\}^{1+m}, \eta', q_{\mathrm{I}}, F)$, where

$$D' := \widehat{D} \cup \{\overline{d}_1 + \overline{d}_1', \ldots, \overline{d}_m + \overline{d}_m'\}\,,$$

and the transition function is defined as follows. Let $p, q \in Q$, $b \in \Sigma$, and $\overline{d} \in D'$, and $d \in \{0,1\}$. For $1 \le i \le m$, let $\overline{e}_i \in \{0,1\}^m$ denote the unit vector that as a 1 in the ith dimension.

- If $\overline{d} = \overline{d}_i + \overline{d}_i'$, for some $1 \le i \le m$, and $\bigcap_{x \ge 1} \eta(p, (b, x\overline{d}, d)) \ne \emptyset$ then

  $$\eta'\big(p, (b, \overline{d}, d, \overline{e}_i)\big) := \bigcap_{x \ge 1} \eta\big(p, (b, x\overline{d}, d)\big) \qquad \text{and} \qquad \eta'\big(p, (b, \overline{d}, d, \overline{c})\big) := \emptyset\,,$$

  where $\overline{c} \ne \overline{e}_i$.
- If $\overline{d} \ne \overline{d}_i + \overline{d}_i'$, for all $1 \le i \le m$, or $\overline{d} = \overline{d}_i + \overline{d}_i'$, for some $1 \le i \le m$, and $\bigcap_{x \ge 1} \eta(p, (b, x\overline{d}, d)) = \emptyset$ then

  $$\eta'\big(p, (b, \overline{d}, d, \overline{0})\big) := \eta\big(p, (b, \overline{d}, d)\big) \qquad \text{and} \qquad \eta'\big(p, (p, \overline{d}, d, \overline{c})\big) := \emptyset\,,$$

  where $\overline{c} \ne \overline{0}$.

The Presburger arithmetic formula φ' is defined as

$$\begin{aligned}\varphi'(z_1, \ldots, z_{2k+1+m}) := \exists x_1 \ldots \exists x_m \big(&\textstyle\bigwedge_{1 \le j \le m} (z_{2k+1+j} = 0 \to x_j = 0) \wedge \\ &z_{2k+1} = 1 \wedge \textstyle\bigwedge_{1 \le i \le k} z_{2i-1} - z_{2i} + \sum_{1 \le j \le m} x_j d_{j(2i-1)} \ge 0\big)\,.\end{aligned}$$

⊣

Theorem 28. *For every Parikh automaton $(\mathcal{A}, \varphi)$ there is a $(0,1)$-reversal-bounded multicounter machine $\mathcal{M}$ with*

$$L(\mathcal{A}, \varphi) = T(\mathcal{M}) .$$

Proof (Sketch). Let $(\mathcal{A}, \varphi)$ be a Parikh automaton of dimension N, where $\mathcal{A} = (Q, \Sigma \times D, \delta, q_{\mathrm{I}}, F)$. Without loss of generality assume that $\flat, \sharp \notin \Sigma$, and $q_{\mathrm{F}} \notin Q$.

We proceed in two steps. First, we define a $(0,0)$-reversal bounded N-counter machine $\mathcal{M}$ such that $w \in L(\mathcal{A})$ iff there is a $(0,0)$-computation

$$(q_{\mathrm{I}}, \flat\Psi(w)\sharp, 1, \overline{0}) \rightsquigarrow \cdots \rightsquigarrow (q_{\mathrm{F}}, \flat\Psi(w)\sharp, |w|+2, \overline{c})$$

and $\Phi(w) = \overline{c}$. In the second step, we extend $\mathcal{M}$ such that $\mathcal{M}$ verifies from the configuration $(q_{\mathrm{F}}, \flat\Psi(w)\sharp, |w|+2, \overline{c})$ whether $\varphi[\overline{c}/\overline{z}]$ holds.

1. Step: Let $\mathcal{M} := (Q \cup \{q_{\mathrm{F}}\}, \Sigma, \eta, q_{\mathrm{I}}, \emptyset, \flat, \sharp)$, where η is defined as

$$\eta(p, b, \overline{\chi}) := \begin{cases} \{(q_{\mathrm{I}}, 1, \overline{0})\} & \text{if } p = q_{\mathrm{I}} \text{ and } b = \flat, \\ \{(q, 1, \overline{d}) \mid q \in \delta\big(p, (b, \overline{d})\big)\} & b \in \Sigma \text{ and } p \in Q, \\ \{(q_{\mathrm{F}}, 0, \overline{0})\} & \text{if } p \in F \text{ and } b = \sharp, \\ \emptyset & \text{otherwise,} \end{cases}$$

for $p \in Q \cup \{q_{\mathrm{F}}\}$, $b \in \Sigma \cup \{\flat, \sharp\}$, and $\overline{\chi} \in \{0,1\}^N$. Note that the transitions above of $\mathcal{M}$ may add a natural number to one of its counters. But it is straightforward to define a transition function that only adds the numbers -1, 0, and $+1$ to a counter by adding new states.

2. Step: Note that Presburger arithmetic admits quantifier elimination. In particular, we can construct for $\varphi(z_1, \dots, z_N)$ an equivalent quantifier-free formula $\psi(z_1, \dots, z_N)$ over the first-order language $<$, $+$, 0, 1, and $\cdot \equiv \cdot \bmod n$, for every $n \geq 2$, where $\cdot \equiv \cdot \bmod n$ expresses congruence modulo the fixed divisor n.

For every atomic formula $t < t'$ and $t \equiv t' \bmod n$ it is tedious but not difficult to extend $\mathcal{M}$'s transition function by adding new states and counters such that there is a $(0,1)$-computation

$$(q_{\mathrm{F}}, \flat\sharp, 2, \overline{c}) \rightsquigarrow \cdots \rightsquigarrow (q'_{\mathrm{F}}, \flat\sharp, 2, \overline{c}')$$

iff $\mathfrak{N} \models t < t'[\overline{c}/\overline{z}]$ and $\mathfrak{N} \models t \equiv t' \bmod n[\overline{c}/\overline{z}]$, respectively, where q'_{F} is a new state. Furthermore, it is also not difficult to extend $\mathcal{M}$'s transition function for negations $\neg\psi$, disjunctions $\psi \vee \psi'$, and conjunctions $\psi \wedge \psi'$. $\dashv$

B Additional Proof Details to §3

Proof (Corollary 12). Let $\psi_{\mathcal{A},\varphi}$ be the WS1S$^{\mathrm{card}}$ formula from Theorem 11, where $(\mathcal{A}, \varphi)$ is a Parikh automaton. Let $\Gamma \subseteq \Sigma \times \mathbb{N}^N$ be the alphabet of $\mathcal{A}$. We

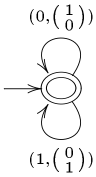

Fig. 4. The automaton that is used in the proof of Lemma 15.

assume without loss of generality that $\Sigma = \{1, \ldots, s\}$, for some $s \geq 1$. Note that the free variables of $\psi_{\mathcal{A}.\varphi}$ are the MSO variables $U_1, \ldots, U_s$. It holds that

$$L(\mathcal{S}) = \Sigma^* \quad \text{iff} \quad (\mathbb{N}, \mathrm{succ}) \models \forall U_1 \ldots \forall U_s \big(\exists n(\psi_{\mathrm{part}}(n, U_1, \ldots, U_s)) \rightarrow \psi_{\mathcal{A},\varphi}(U_1, \ldots, U_s)\big)\,.$$

This sentence can be easily transformed into a sentence of the form $\forall\overline{U}\exists\overline{X}\chi$. The claim follows from the undecidability of the universal problem for Parikh automata (Theorem 7). ⊣

Proof (Lemma 14). Let $QX_1 \ldots QX_n\varphi$ be a formula in $\mathsf{F}_\exists \cup \mathsf{F}_\forall$. First, we can assume that the connective $\neg$ only appears directly in front of the atomic formulas. When φ is already a Boolean combination of WS1S formulas and formulas in C then we are done. Assume that there is a subformula in φ of the form $Q'\alpha(\psi \oplus \psi')$, where $Q' \in \{\exists, \forall\}$, $\oplus \in \{\vee, \wedge\}$, $\psi \in \mathsf{C}$, and ψ' is a WS1S. This formula is logically equivalent to $\psi \oplus Q'\alpha\psi'$ since α does not occur in ψ. ⊣

Proof (Lemma 15). We need the following notation. The number of occurrences of a symbol b in word w is denoted by $\#_b(w)$.

Assume first that $\psi = c + |X_1| + \cdots + |X_r| < d + |Y_1| + \cdots + |Y_s|$. Let $\tau : (\{0,1\}^k)^* \rightarrow \{0,1\}^*$ be the homomorphism given by

$$b \overset{\tau}{\mapsto} 0^u \quad \text{if } u \geq 0\,, \qquad \text{and} \qquad b \overset{\tau}{\mapsto} 1^{-u} \quad \text{if } u < 0\,,$$

where $u := \sum_{1 \leq j \leq r} \chi_{X_j}(b) - \sum_{1 \leq j \leq s} \chi_{Y_j}(b)$. It is easy to check that

$$w \in L(\psi) \qquad \text{iff} \qquad c - d + \#_0(\tau(w)) < \#_1(\tau(w))\,.$$

The deterministic Parikh automata $(\mathcal{A}, c + z_1 < d + z_2)$, where $\mathcal{A}$ is depicted in Figure 4 recognizes the language $\{w \in \{0,1\}^* \mid c - d + \#_0(w) < \#_1(w)\}$. By Theorem 19 there is a deterministic Parikh automaton recognizing $L(\psi)$.

The case $\neg\psi$ follows now from Theorem 23. ⊣

C Additional Details to §4

We recall the definition of tree automata. A **(top-down) tree automaton** $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, q_{\mathrm{I}}, F)$, where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q \times Q)$ is the transition function, q_{I} is the initial state, and

$F \subseteq Q$ is the set of final states. A **run** ϱ of $\mathcal{A}$ on $t \in T_\Sigma$ is a Q-labeled tree ϱ with $\mathrm{dom}(\varrho) = \{\lambda\} \cup \{ub \mid u \in \mathrm{dom}(t) \text{ and } b \in \{0,1\}\}$ such that $\varrho(u) = q_\mathrm{I}$, and $(\varrho(u0), \varrho(u1)) \in \delta(\varrho(u), t(u))$, for all $u \in \mathrm{dom}(t)$. ϱ is **accepting** if all leaves of ϱ are labeled with states in F, i.e. $\varrho(u) \in F$, for all $u \in \mathrm{dom}(\varrho) \setminus \mathrm{dom}(t)$. A tree t is **recognized** by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on t. $T(\mathcal{A})$ denotes the set of trees that are recognized by $\mathcal{A}$.

We formally define Parikh tree automata.

Definition 29. *A **Parikh tree automaton** of dimension $N \geq 0$ is a pair $(\mathcal{A}, \varphi)$, where $\mathcal{A}$ is a tree automaton with the alphabet $\Sigma \times D$, where $D \subseteq \mathbb{N}^N$, and $\varphi(z_1, \dots, z_N)$ is a Presburger arithmetic formula. $(\mathcal{A}, \varphi)$ **recognizes** the language $T(\mathcal{A}, \varphi) := T(\mathcal{A}) \restriction_\varphi$.*

*A Parikh tree automaton $(\mathcal{A}, \varphi)$ is **deterministic** if for the transition function δ of $\mathcal{A}$ holds that for every state q and for every $(b, \overline{d}) \in \Sigma \times D$, $|\delta(q, (b, \overline{d}))| \leq 1$, and if $|\delta(q, (b, \overline{d}))| = 1$ then $|\delta(q, (b, \overline{d}'))| = 0$, for every $\overline{d} \neq \overline{d}'$.*

C.1 Some Properties about Parikh Tree Automata

The proof of Lemma 5 carries over to regular tree languages. Therefore, we obtain the following lemma.

Lemma 30. *Let Γ be an alphabet of the form $\Sigma \times D$ where $D \subseteq \mathbb{N}^n$, for some $n \geq 0$. For every regular tree language $L \subseteq T_\Gamma$, there exist linear polynomials $q_1, \dots, q_m$ such that*

$$\Phi(L) = \bigcup_{1 \leq i \leq m} \{q_i(x_1, \dots, x_r) \mid x_1, \dots, x_r \in \mathbb{N}\},$$

where Φ is the Parikh map of Γ. Moreover, the polynomials $q_1, \dots, q_m$ are effectively constructible.

We conclude now, as for Parikh automata, from the lemma above the decidability of the emptiness problem for Parikh tree automata.

Property 31. *The emptiness problem for Parikh tree automata is decidable.*

It is straightforward to reduce the universality problem for Parikh automata to Parikh tree automata.

Property 32. *The universality problem for Parikh tree automata is undecidable.*

The following closure properties of Parikh tree automata can be proved analogously to the closure properties of Parikh automata in A.1.

Property 33. *(1) Parikh tree automata are closed under union, intersection, homomorphisms, and inverse homomorphisms, but not under complement.*
(2) Deterministic Parikh tree automata are closed under union, intersection, complement, and inverse homomorphisms, but not under homomorphisms.

C.2 WS2S with Cardinality Constraints

For a detailed account on WS2S see, e.g., Thomas' survey article [30]. We adopt the notations and definitions we have introduced in §3 for WS1S and $\mathrm{WS1S}^{\mathrm{card}}$ to WS2S and $\mathrm{WS2S}^{\mathrm{card}}$.

Analogously to words over the alphabet $\{0,1\}^k$, a tree $t \in T_{\{0,1\}^k}$ determines the interpretation I_t for variables $\alpha_1, \ldots, \alpha_k$, where for all $1 \leq j \leq k$,

- $u = I_t(\alpha_j)$ iff $\chi_j(t(u)) = 1$ and $\chi_j(t(u')) = 0$ for all $u' \neq u$, whenever α_i is a FO variable, and
- $u \in I_t(\alpha_j)$ iff $\chi_j(t(u)) = 1$, whenever α_j is a MSO variable.

We extend χ_j homomorphically to trees. For a formula $\varphi(\alpha_1, \ldots, \alpha_k)$, we define

$$T(\varphi) := \{t \in T_{\{0,1\}^k} \mid (\mathbb{T}, \mathrm{succ}_0, \mathrm{succ}_1), I_t \models \varphi\}\,.$$

As for WS1S, the language described by some WS2S formula is a regular tree language [9, 29].

Fact 34. $T(\varphi)$ is regular for every WS2S formula φ. Moreover, we can effectively build a tree automaton recognizing $T(\varphi)$.

Before we launch into the proof of Theorem 18 we need the following lemmas that can be proved analogously to their analogues Lemma 14 and Lemma 15, respectively.

Lemma 35. *Let $Q_1 x_1.(\psi_{\mathrm{FO}}(x_1)) \ldots Q_m x_m.(\psi_{\mathrm{FO}}(x_m))\, QX_1 \ldots QX_n \varphi$ be a $\mathrm{WS2S}^{\mathrm{card}}$ formula, where $Q_1, \ldots, Q_m, Q \in \{\exists, \forall\}$ and all MSO variables occurring in a subformula in C of φ are bounded by one of the quantifiers for $X_1, \ldots, X_n$. Then, the formula φ can be effectively transformed into a logical equivalent formula of the form $\bigvee_i \bigwedge_j \psi_{ij}$, where each ψ_{ij} is either a WS2S formula or $\psi_{ij} \in \mathsf{C}$.*

Lemma 36. *For every $\psi \in \mathsf{C}$ there is a Parikh tree automaton recognizing $T(\psi)$.*

The proof of Theorem 18 follows the lines of the proof of Theorem 16.

Proof (Theorem 18). Let θ be a sentence

$$Q_1 x_1.(\psi_{\mathrm{FO}}(x_1)) \ldots Q_m x_m.(\psi_{\mathrm{FO}}(x_m))\, QX_1 \ldots QX_n \varphi\,,$$

where $Q_1, \ldots, Q_m, Q \in \{\exists, \forall\}$ and all MSO variables occurring in a subformula in C of φ are bounded by one of the quantifiers for $X_1, \ldots, X_n$.

<u>Case I:</u> $Q = \exists$. Because of Lemma 35, we can assume that $\varphi = \bigvee_i \bigwedge_j \psi_{ij}$, where ψ_{ij} is either a WS2S formula or $\psi_{ij} \in \mathsf{C}$.

We can construct a tree automaton $\mathcal{A}_{ij}$ with $T(\mathcal{A}_{ij}) = T(\psi_{ij})$ if ψ_{ij} is a WS2S formula, and we can construct a deterministic Parikh tree automaton $(A_{ij}, \widehat{\psi}_{ij})$ such that $T(\mathcal{A}_{ij}, \widehat{\psi}_{ij}) = T(\psi_{ij})$ because of Lemma 36. Since Parikh tree automata are closed under union and intersection, we can construct a Parikh tree automaton $(\mathcal{A}, \widehat{\varphi})$ with $T(\mathcal{A}, \widehat{\varphi}) = T(\varphi)$. Assume that the dimension of $(\mathcal{A}, \varphi)$ is $N \geq 0$ and that $\mathcal{A} = (S, \Gamma, \delta, s_{\mathrm{I}}, F)$, where $\Gamma \subseteq \{0,1\}^{m+n} \times \mathbb{N}^N$.

It holds that $(\mathbb{T}, \mathrm{succ}_0, \mathrm{succ}_1) \models \theta$ is equivalent to

$$\begin{aligned} &Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there exists a tree } t \in T(\mathcal{A}, \widehat{\varphi}) \text{ such that} \\ &\qquad I_t(x_1) = 0^{\tilde{x}_1}, \ \ldots, \ I_t(x_m) = 0^{\tilde{x}_m} . \end{aligned} \tag{6}$$

By definition, (6) is equivalent to

$$\begin{aligned} &Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there exists a tree } t \in T(\mathcal{A}) \text{ such that} \\ &\qquad I_{\Psi(t)}(x_1) = 0^{\tilde{x}_1}, \ \ldots, \ I_{\Psi(t)}(x_m) = 0^{\tilde{x}_m} \text{ and } \mathfrak{N} \models \widehat{\varphi}[\Phi(t)/\overline{z}] , \end{aligned} \tag{7}$$

where Ψ is the projection from Γ to $\{0,1\}^{m+n}$, and Φ is the Parikh map of Γ.

We extend the alphabet Γ to $\Gamma' := \{(b, \overline{v}, \overline{v'}) \mid (b, \overline{v}) \in \Gamma \text{ and } \overline{v}' \in \{0,1\}^m\}$, i.e., we append the vectors $\overline{v}' \in \{0,1\}^m$ to each symbol $(b, \overline{v}) \in \Gamma$. Let Φ' be the Parikh map of Γ'. In the following, we construct a tree automaton $\mathcal{A}'$ with alphabet Γ' such that

$$\begin{aligned} T(\mathcal{A}') = \{t \in T_{\Gamma'} \mid \ & h(t) \in T(\mathcal{A}), t(u) \in \Gamma \times \{0\}^m, \text{ for all } u \in \mathrm{dom}(t) \setminus \{0\}^*, \text{ and} \\ & \Phi'(t) = \big(\Phi(h(t)), |I_{\Psi(h(t))}(x_1)|, \ldots, |I_{\Psi(h(t))}(x_m)|\big)\} , \end{aligned}$$

where $h : \Gamma'^* \to \Gamma^*$ is the homomorphism defined by $h(b, \overline{v}, \overline{v}') := (b, \overline{v})$.

Let $\mathcal{B} := (S, \Gamma', \delta', s_{\mathrm{I}}, F)$, where the transition function δ' contains the same transitions as δ except that $\mathcal{B}$ marks the lengths of the nodes in a tree that determine of the interpretations for the FO variables $x_1, \ldots, x_m$. For each FO variable x_i, let $B_i \subseteq S$ be the set that contains all the states that are reachable *before* reading a symbol that determines the interpretation of the FO variable x_i, i.e. $B_i := \bigcup_{j \in \mathbb{N}} B_i^j$, where $B_i^0 := \{s_{\mathrm{I}}\}$ and for $j > 0$,

$$B_i^{j+1} := \{s_0 \in S \mid (s_0, s_1) \in \delta\big(B_i^j, (b, \overline{v})\big) \text{ with } \chi_{x_i}(b) = 0\} .$$

For $s \in S$, let $\overline{c}(s) \in \{0,1\}^m$ be the characteristic vector of s, i.e. $\overline{c}(s) := (c_1, \ldots, c_m)$, where $c_i = 1$ iff $s \in B_i$. Now, $\delta' : S \times \Gamma' \to \mathcal{P}(S)$ is defined by

$$\delta'(s, (b, \overline{v}, \overline{v}')) := \{(s_0, s_1) \in \delta(s, (b, \overline{v})) \mid \overline{c}(s_0) = \overline{v}'\} .$$

From $\mathcal{B}$ we obtain $\mathcal{A}'$ by constructing the tree automaton that recognizes the intersection of $T(\mathcal{B})$ and the regular set $\{t \in T_{\Gamma'} \mid t(u) \in \Gamma \times \{0\}^m, \text{ for all } u \in \mathrm{dom}(t) \setminus \{0\}^*\}$.

By the construction of $\mathcal{A}'$, (7) is equivalent to

$$\begin{aligned} &Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there exists a tree } t \in T(\mathcal{A}') \text{ such that} \\ &\qquad \mathfrak{N} \models \widehat{\varphi}[\Phi(h(t))/\overline{z}] \text{ and } \Phi'(t) = (\Phi(h(t)), \tilde{x}_1, \ldots, \tilde{x}_m) . \end{aligned} \tag{8}$$

From Lemma 30, it follows that there are polynomials $q_1, \ldots, q_k$ such that $\Phi'(T(\mathcal{A}'))$ is the union of the images of the polynomials $q_1, \ldots, q_k$. We conclude that (8) is equivalent to

$$\begin{aligned} &Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there exist } y_1, \ldots, y_r \in \mathbb{N} \text{ and } \overline{v} \in \mathbb{N}^N \\ &\qquad \text{such that } \mathfrak{N} \models \widehat{\varphi}[\overline{v}/\overline{z}] \text{ and} \\ &\qquad q_i(y_1, \ldots, y_r) = (\overline{v}, \tilde{x}_1, \ldots, \tilde{x}_m), \text{ for some } 1 \leq i \leq k . \end{aligned} \tag{9}$$

Note that (9) can be expressed as a sentence in Presburger arithmetic. Since Presburger arithmetic is decidable we are done.

Case II: $Q = \forall$. Follows from Case I, by the duality of quantifiers. ⊣