Monadic Second-Order Logics with Cardinalities*

Felix Klaedtke[1] and Harald Rueß[2]

[1] Albert-Ludwigs-Universität Freiburg, Germany
[2] SRI International, CA, USA

Abstract. We delimit the boundary between decidability versus undecidability of the weak monadic second-order logic of one successor (WS1S) extended with linear cardinality constraints of the form $|X_1|+\cdots+|X_r| < |Y_1|+\cdots+|Y_s|$, where the X_is and Y_js range over finite subsets of natural numbers. Our decidability and undecidability results are based on an extension of the classic logic-automata connection using a novel automaton model based on Parikh maps.

1 Introduction

In the automata-theoretic approach for solving the satisfiability problem of a logic one develops an appropriate notion of automata and establishes a translation from formulas to automata. The satisfiability problem for the logic then reduces to the automata emptiness problem. Most prominently, decidability of the (weak) monadic second-order logic of one successor (W)S1S is proved by a translation of formulas to word automata, see e.g. [27]. Despite the nonelementary worst-case complexity [19, 26], the automata-based decision procedure for WS1S, implemented in the Mona tool [10, 16], has been found to be effective for reasoning about a multitude of computation systems ranging from circuits [3, 2] to protocols [17, 25]. Furthermore, it has been integrated in theorem provers to decide well-defined fragments of higher-order logic [1, 21].

Many interesting verification problems, however, fall outside the scope of WS1S. For example, the verifications in WS1S for the sequential circuits considered in [3] are only with respect to concrete values of parameters such as setup time and minimum clock period since some linear arithmetic is used on these parameters. Also, certain distributed algorithms such as the Byzantine generals problem [18] of reaching distributed consensus in the presence of unreliable messengers and treacherous generals cannot be modeled in WS1S, since reasoning about the combination of (finite) sets and cardinality constraints on these sets is required here.

In order to support this kind of reasoning and to significantly extend the range of automated verification procedures we extend WS1S with atomic formulas of the form $|X_1| + \cdots + |X_r| < |Y_1| + \cdots + |Y_s|$, where the X_is and Y_js are

* This work was supported by SRI International internal research and development, and NASA through contract NAS1-00079.

J.C.M. Baeten et al. (Eds.): ICALP 2003, LNCS 2719, pp. 681–696, 2003.

monadic second-order (MSO) variables, and $|X|$ denotes the cardinality of the MSO variable X. The extension of WS1S with cardinality constraints is denoted by $\text{WS1S}^{\text{card}}$. Our main results are

(i) $\text{WS1S}^{\text{card}}$ is undecidable. More precisely, (a) the fragment of $\text{WS1S}^{\text{card}}$ consisting of the sentences of the form $\forall X \exists \overline{Y} \varphi$ is undecidable, where X is an MSO variable, $\overline{Y}$ is a vector of MSO variables ranging over finite sets of natural numbers, and all quantifiers in φ are first-order, that is, ranging over natural numbers. And, (b) the fragment of $\text{WS1S}^{\text{card}}$ consisting of the sentences of the form $\exists X \forall y \exists \overline{Y} \varphi$, where y is a first-order variable and all quantifiers in φ are first-order.

(ii) The fragment consisting of the sentences of the form $Q_1 x_1 \dots Q_\ell x_\ell Q \overline{Y} \varphi$ is decidable, where $Q_1, \dots, Q_\ell \in \{\exists, \forall\}$ are first-order quantifiers and an MSO variable occurring in a cardinality constraint in φ is bound by $Q \in \{\exists, \forall\}$.

Together the results (i) and (ii) delimit the boundary between decidability versus undecidability of MSO logics with cardinality constraints.

We use an automata-theoretic approach for obtaining these results by defining a suitable extension of finite word automata. These extensions work over an extended alphabet in which a vector of natural numbers is attached to each letter of the input alphabet. An input is accepted if an input word is accepted in the traditional sense *and* if a projection of the word via a monoid homomorphism to a vector of natural numbers satisfies given arithmetic constraints. Since this monoid homomorphism generalizes Parikh's *commutative image* [23] on words, we call such an extended automaton a *Parikh finite word automaton* (PFWA). PFWAs characterize the expressiveness of the existential fragment of $\text{WS1S}^{\text{card}}$.

The undecidability results (i) follow from the undecidability of the universality problem for PFWAs and the undecidability of the halting problem for 2-register machines, whereas the decidability result (ii) is based on a two-step construction. First, we build a PFWA for the formula $\exists \overline{Y} \varphi$, and, second, we transform the PFWA into a corresponding Presburger arithmetic formula. This latter construction takes care of the quantification of the first-order variables $x_1, \dots, x_\ell$. Compared to simply checking the emptiness problem for the PFWA associated with a formula, this two-stage translation yields a decision procedure for a much more expressive fragment of $\text{WS1S}^{\text{card}}$. These constructions can readily be extended to obtain corresponding results for cardinality constraints in second-order monadic logics over trees [15].

The paper is structured as follows. In §2 we introduce PFWAs. Then, in §3 we define $\text{WS1S}^{\text{card}}$ and compare the expressiveness of the existential fragment of $\text{WS1S}^{\text{card}}$ with PFWAs. In §4 we prove the results (i) and (ii), and illustrate applications of the decidability result (ii). Finally, in §5 we draw conclusions.

2 Parikh Automata

We introduce a framework that extends the acceptance condition of machines operating on words. In addition to the traditional acceptance condition of a

machine, we require that an input satisfies arithmetic properties, where the input is associated with a vector of natural numbers. Parikh finite word automata are an instance of this framework.

Let $\Sigma = \{b_1, \ldots, b_n\}$ be a linearly ordered alphabet. Parikh's [23] *commutative image* $\Phi : \Sigma^* \to \mathbb{N}^{|\Sigma|}$ maps the elements of the free monoid Σ^*, so-called words, to vectors of natural numbers. The commutative image is defined by $\Phi(b_i) := \overline{e}_i$ and $\Phi(uv) := \Phi(u) + \Phi(v)$, where $\overline{e}_i \in \mathbb{N}^{|\Sigma|}$ is the unit vector with the ith coordinate equal to 1 and all other coordinates are 0. Intuitively, the ith position of $\Phi(w)$ counts how often b_i occurs in $w \in \Sigma^*$. We extend Parikh's commutative image by considering the Cartesian product of Σ and a nonempty set D of vectors of natural numbers.

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

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

Note that if we attach to each letter $b_i \in \Sigma$ the unit vector $\overline{e}_i \in \mathbb{N}^{|\Sigma|}$ in a word $w \in \Sigma^*$ then the extended Parikh map yields the commutative image of w.

We constrain a language by an arithmetic property given by a set of vectors of natural numbers.

Definition 2. *For a language $L \subseteq (\Sigma \times D)^*$ and $C \subseteq \mathbb{N}^N$, let*

$$L \restriction_C := \{\Psi(w) \mid w \in L \text{ and } \Phi(w) \in C\}$$

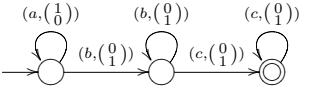
be the ***restriction*** *of L with respect to C.*

The acceptance condition of a machine operating on words can be extended in the following way. A word w over Σ is accepted if the machine accepts a word over $\Sigma \times D$ both in the traditional sense and if the sum of the attached vectors to the symbols in w is in a given subset of $\mathbb{N}^N$. Here, we are mainly concerned with finite word automata and arithmetic constraints restricted to *semilinear sets* $U \subseteq \mathbb{N}^s$. This means that there are linear polynomials $p_1, \ldots, p_m : \mathbb{N}^r \to \mathbb{N}^s$ such that U is the union of the images of these polynomials, that is, $U = \bigcup_{1 \leq i \leq m} \{p_i(x_1, \ldots, x_r) \mid x_1, \ldots, x_r \in \mathbb{N}\}$.

Definition 3. *A* ***Parikh finite word automaton (PFWA)*** *of dimension $N \geq 1$ is a pair $(\mathcal{A}, C)$, where $\mathcal{A}$ is a finite word automaton with an alphabet of the form $\Sigma \times D$, D is a finite, nonempty subset of $\mathbb{N}^N$, and C is a semilinear subset of $\mathbb{N}^N$. The PFWA $(\mathcal{A}, C)$* ***recognizes*** *the language $L(\mathcal{A}, C) := L(\mathcal{A}) \restriction_C$, where $L(\mathcal{A})$ is the language recognized by $\mathcal{A}$.*

The PFWA $(\mathcal{A}, C)$ 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 PFWA $(\mathcal{A}, \{\binom{z}{z} \mid z \in \mathbb{N}\})$, where $\mathcal{A}$ is given by the picture



recognizes $\{a^{i+j}b^ic^j \mid i,j>0\}$, which is context-sensitive but not context-free.

PFWAs are strictly more expressive than finite word automata, where the accepted words are constrained by their commutative images and semilinear sets. It is easy to define a deterministic PFWA automaton that recognizes the language $L := \{a^ib^ja^ib^j \mid i,j \geq 1\}$. But there does not exist a finite word automaton $\mathcal{A}$ with the alphabet $\{a,b\}$ and a set $C \subseteq \mathbb{N}^2$ such that $w \in L$ iff $w \in L(\mathcal{A})$ and $(k,\ell) \in C$, where (k,ℓ) is the commutative image of w.

A PFWA can be seen as a finite word automaton 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 automaton models in the literature, for example [4,7,11], we do not restrict the applicability of transitions in a run by additional guards on the values of the counters. Instead, a PFWA constrains the language of a finite word automaton over the extended alphabet by a semilinear set. It turns out (a) that PFWAs are equivalent to reversal-bounded multicounter machines [11,12] and (b) that PFWAs are equivalent to weighted finite automata over the groups $(\mathbb{Z}^k,+,0)$ [6,20] with $k \geq 1$ in the sense that all these three different kinds of machines describe the same class of languages. We refer the reader to [15], for definitions and a detailed comparison of these automaton models, proofs of the equivalences, and a comparison of PFWAs to other automaton models.

We state some properties of PFWAs. The details can be found in [15].

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

The decidability of the emptiness problem relies on Parikh's result [23], which states that the commutative image of a context-free language is semilinear.

Lemma 5. *Let Γ be an alphabet of the form $\Sigma \times D$ with $D \subseteq \mathbb{N}^N$, for some $N \geq 1$. For every context-free language $L \subseteq \Gamma^*$, there are linear polynomials $q_1,\ldots,q_m : \mathbb{N}^r \to \mathbb{N}^N$, for some $r \geq 1$, 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 extended Parikh map of Γ. Moreover, the polynomials $q_1,\ldots,q_m$ are effectively constructible if L is given by a pushdown automaton.

For a PFWA $(\mathcal{A},C)$, we know by Lemma 5 that the set $\Phi(L(\mathcal{A}))$ is semilinear and effectively constructible. The decidability of the emptiness problem follows from the facts that semilinear sets are effectively closed under intersection, and $L(\mathcal{A},C) \neq \emptyset$ iff $\Phi(L(\mathcal{A})) \cap C \neq \emptyset$.

Property 6. *The emptiness problem for PFWAs is decidable.*

The undecidability of the universality problem for PFWAs can be shown by reduction from the word problem for Turing machines.

Property 7. *The universality problem for PFWAs is undecidable.*

Note that the universality problem for deterministic PFWAs is decidable since they are closed under complement and the emptiness problem is decidable.

3 WS1S with Cardinality Constraints

We extend WS1S in order to compare cardinalities of sets. We call this extension $\mathrm{WS1S}^{\mathrm{card}}$. The classic logic-automata connection of finite word automata and WS1S extends to PFWAs and the existential fragment of $\mathrm{WS1S}^{\mathrm{card}}$.

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. We adopt the following notation: lowercase letters $x, y, \ldots$ denote FO variables and uppercase 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, and we use the standard conventions for omitting parentheses. A formula is **existential** if it is of the form $\exists \overline{X} \varphi$ where all bound variables in φ are FO.

Formulas are interpreted over the natural numbers with the successor relation, that is, the structure $(\mathbb{N}, \mathrm{succ})$. An 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 in the obvious way. 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, that is, φ does not have free variables.

Equality $x = y$ can be expressed by $\exists z\big(\mathrm{succ}(x, z) \wedge \mathrm{succ}(y, z)\big)$. For a natural number $t \in \mathbb{N}$, we write $x = t$ for $\exists z_0 \ldots \exists z_t \big(x = z_t \wedge \bigwedge_{0 \leq i < t} \mathrm{succ}(z_i, z_{i+1}) \wedge \forall y \neg \mathrm{succ}(y, z_0)\big)$, and we write $Yx + 1$ for $\exists z\big(\mathrm{succ}(x, z) \wedge Yz\big)$. The formula $\mathit{part}(U, X_1, \ldots, X_k)$ expresses that $X_1, \ldots, X_k$ is a partition of U, that is $\forall y\big(Uy \leftrightarrow \bigvee_{1 \leq i \leq k} X_i y\big) \wedge \neg \exists y\big(\bigvee_{1 \leq i < j \leq k} (X_i y \wedge X_j y)\big)$. Note that in these formulas only FO variables are quantified.

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 $1 \leq j \leq k$,

- $i \in I_w(\alpha_j)$ iff $\chi_j(b_i) = 1$, if α_j is an MSO variable, and
- $i = I_w(\alpha_j)$ iff $\chi_j(b_i) = 1$ and $\chi_j(b_{i'}) = 0$ for all $i' \neq i$, if α_i is an FO 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 $b_1 \ldots b_n$, that is, $\chi_j(b_1 \ldots b_n)$. 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 facts that are due to Büchi, Elgot, and Trakhtenbrot. For more details, see, for example, [27].

Fact 8. $L(\varphi)$ is regular for every WS1S formula φ. Moreover, we can effectively build a finite word automaton recognizing $L(\varphi)$.

For the other direction, that is, describing regular languages by WS1S formulas, there is a subtlety that we want to point out. Note that natural numbers and finite subsets of $\mathbb{N}$ have several encodings, e.g., all the words in $\{0\}^*$ encode the empty set. It is easy to see that languages definable by WS1S formulas, are closed under $\overline{0}$-padding and $\overline{0}$-cutting, that is, $w \in L(\varphi)$ iff $w\overline{0} \in L(\varphi)$, where $\overline{0}$ is the letter $(0, \ldots, 0)$. We call a $\overline{0}$-padding and $\overline{0}$-cutting closed language **$\overline{0}$-closed**.

Fact 9. For every regular $\overline{0}$-closed language $L \subseteq (\{0,1\}^k)^*$ there is an existential WS1S formula $\varphi(X_1, \ldots, X_k)$ such that $L(\varphi) = L$.

To obtain an equivalence of the logic and the regular languages, one has to look at finite *word models* [27]. The main difference is that the universe of a finite word model is not $\mathbb{N}$, but $\{0, \ldots, n-1\}$ where n is given by the length of the word. The distinction between the different semantics is emphasized by using the name M2L(str) or MSO[+1] instead of WS1S. The results below carry over to finite word models. We use the WS1S semantics since it simplifies matters.

Cardinality Constraints. $\mathrm{WS1S}^{\mathrm{card}}$ has in addition to the atomic formulas of WS1S the atomic formulas of the form $|X_1| + \cdots + |X_r| < |Y_1| + \cdots + |Y_s|$, where the truth value with respect to an interpretation I is defined as

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

Let C be the set of formulas of the form $|X_1|+\cdots+|X_r| < |Y_1|+\cdots+|Y_s|$ and their negations. We write formulas like $-2|X| = 3|Y| + |Z|$ which can be transformed to an equivalent Boolean combination of formulas in C by standard arithmetic. Moreover, we also use the summation symbol $\sum$ for a shorter representation.

Parikh Automata and $\mathrm{WS1S}^{\mathrm{card}}$. We carry over the Facts 8 and 9 to the existential fragment of $\mathrm{WS1S}^{\mathrm{card}}$ and PFWAs. We start with the direction in Fact 9.

Theorem 10. *For every PFWA $(\mathcal{A}, C)$ where $L(\mathcal{A}, C) \subseteq (\{0,1\}^s)^*$ is $\overline{0}$-closed, there is an existential* $\mathrm{WS1S}^{\mathrm{card}}$ *formula $\psi_{\mathcal{A},C}(U_1, \ldots, U_s)$ with $L(\varphi) = L(\mathcal{A}, C)$.*

Proof. Let $N \geq 1$ be the dimension of $(\mathcal{A}, C)$, and let $\mathcal{A} = (Q, \{0,1\}^s \times D, \delta, q_{\mathrm{I}}, F)$. Without loss of generality, we assume that $Q = \{1, \ldots, r\}$, for some $r \geq 1$. Let K be the maximal natural number occurring in a vector in D, that is $K := \max\big(\bigcup_{(d_1,\ldots,d_N)\in D}\{d_1, \ldots, d_N\}\big)$.

Let $b_0 \ldots b_{n-1} \in L(\mathcal{A}, C)$ with $n \geq 0$ and $b_{n-1} \neq \overline{0}$. The formula $\psi_{\mathcal{A},C}$ describes the existence of an accepting run $\varrho = q_0 \ldots q_{n+m} \in Q^*$ on a word $(b_0, \overline{d}_0) \ldots (b_{n-1}, \overline{d}_{n-1})(\overline{0}, \overline{d}_n) \ldots (\overline{0}, \overline{d}_{n+m-1}) \in (\{0,1\}^s \times D)^*$, for some $m \geq 0$. Note that $L(\mathcal{A}, C)$ is $\overline{0}$-closed. It holds, $q_0 = q_{\mathrm{I}}$, $q_{i+1} \in \delta(q_i, (b_i, \overline{d}_i))$, for $0 \leq i < n+m$, and $q_{n+m} \in F$. We encode ϱ by pairwise disjoint sets $Y_1, \ldots, Y_r \subseteq \{0, \ldots, n+m\}$ such that Y_q contains those positions i with $q = q_i$. Moreover, we keep track of the numbers at the kth position of the vectors $\overline{d}_i$ with the sets $Z_k^0, \ldots, Z_k^K \subseteq \{0, \ldots, n+m\}$: it holds that $0 \in Z_k^0$ and $i \in Z_k^d$ iff the kth position of $\overline{d}_{i-1}$ is d, for $1 \leq i \leq n+m$. Therefore, the kth position of the vector $\overline{d}_0 + \cdots + \overline{d}_{n+m-1}$ is $\sum_{0\leq d\leq K} d|Z_k^d|$. We have to check $\overline{d}_0 + \cdots + \overline{d}_{n+m-1} \in C$. Formally,

$$
\begin{aligned}
\exists Y_1 \ldots \exists Y_r \exists Z_1^0 \ldots \exists Z_1^K \ldots \exists Z_N^0 \ldots \exists Z_N^K \exists U \Big(& \mathit{domain}(U, U_1, \ldots, U_s) \wedge \\
& \mathit{part}(U, Y_1, \ldots, Y_r) \wedge \textstyle\bigwedge_{1\leq i\leq N} \mathit{part}(U, Z_i^0, \ldots, Z_i^K) \wedge \\
& \forall x \Big(\big(x = 0 \to Y_{q_{\mathrm{I}}} x \wedge \textstyle\bigwedge_{1\leq i\leq N} Z_i^0 x\big) \wedge \\
& \quad \big(Ux \to \textstyle\bigvee_{q\in\delta(p,(b,(d_1,\ldots,d_N)))} (Y_p x \wedge \mathit{letter}_b(x, U_1, \ldots, U_s) \wedge \\
& \qquad\qquad Y_q x + 1 \wedge \textstyle\bigwedge_{1\leq i\leq N} Z_i^{d_i} x + 1)\big) \wedge \\
& \quad \big(Ux \wedge \neg Ux + 1 \to \big(\textstyle\bigvee_{q\in F} Y_q x\big)\big)\Big) \wedge \\
& \psi_C(Z_1^0, \ldots, Z_1^K, \ldots, Z_N^0, \ldots, Z_N^K)\Big),
\end{aligned}
$$

where $\mathit{domain}(U, U_1, \ldots, U_s)$ is the formula $\forall x(U_1 x \vee \cdots \vee U_s x \to Ux + 1) \wedge \forall x(Ux + 1 \to Ux)$ and $\mathit{letter}_b(x, U_1, \ldots, Us) := (\bigwedge_{b_i=0} \neg U_i x) \wedge (\bigwedge_{b_i=1} U_i x)$, for $b = (b_1, \ldots, b_s) \in \{0,1\}^s$. It remains to define the formula ψ_C. Since C is semilinear we can assume that C is the union of the images of linear polynomials $p_1, \ldots, p_\ell : \mathbb{N}^k \to \mathbb{N}^N$, for some $k \geq 1$. For $1 \leq i \leq \ell$, let

$$
\begin{aligned}
\psi_{p_i} := \exists X_1 \ldots \exists X_k \Big(\textstyle\bigwedge_{1\leq j\leq N} \exists X \big(\forall y(Xy \leftrightarrow \bigvee_{0\leq t<d_0^j} y = t) \wedge \\
\textstyle\sum_{1\leq d\leq K} d|Z_j^d| = |X| + d_1^j|X_1| + \cdots + d_k^j|X_k|\big)\Big),
\end{aligned}
$$

where we assume that $p_i(x_1, \ldots, x_k)$ has the form $\overline{d}_0 + \overline{d}_1 x_1 + \cdots + \overline{d}_k x_k$ with $\overline{d}_j = (d_j^1, \ldots, d_j^N)$, for $0 \leq j \leq k$. Let $\psi_C := \psi_{p_1} \vee \cdots \vee \psi_{p_\ell}$. The quantification of the MSO variables in ψ_C can be pulled out to obtain an existential formula. ⊣

We give a set of formulas describing languages recognizable by PFWAs. This carries over Fact 8 to PFWAs and the existential fragment of $\mathrm{WS1S}^{\mathrm{card}}$. Let E be the set of $\mathrm{WS1S}^{\mathrm{card}}$ formulas of the form $\exists X_1 \ldots \exists X_n \varphi$, where all MSO variables occurring in a subformula in C of φ are either free or bound by one of the existential quantifiers for $X_1, \ldots, X_n$. Note that an existential formula

is in E, but E contains formulas that are not existential. A formula in E can contain MSO variables Y that are universally quantified if Y does not occur in subformulas in C and the quantification of Y happens below the existential quantification of $X_1, \ldots, X_n$.

Theorem 11. *For every $\varphi \in \mathsf{E}$, we can construct a PFWA recognizing $L(\varphi)$.*

Proof (Sketch). We can assume that $\varphi \in \mathsf{E}$ is of the form $\exists X_1 \ldots \exists X_n (\bigvee_i \bigwedge_j \psi_{ij})$, where ψ_{ij} is either a WS1S formula or $\psi_{ij} \in \mathsf{C}$. By Fact 8 we can construct a finite word automaton $\mathcal{A}_{ij}$ with $L(\mathcal{A}_{ij}) = L(\psi_{ij})$ if ψ_{ij} is a WS1S formula, and for $\psi_{ij} \in \mathsf{C}$, it is straightforward to give a PFWA (A_{ij}, C_{ij}) with $L(\mathcal{A}_{ij}, C_{ij}) = L(\psi_{ij})$. From Property 4 we can construct a PFWA recognizing $L(\varphi)$. ⊣

Theorems 10 and 11 together reveal the following equivalence.

Corollary 12. *For a $\overline{0}$-closed language $L \subseteq (\{0,1\}^s)^*$, the following two conditions are equivalent:*

(i) L is recognizable by a PFWA, that is, there is a PFWA $(\mathcal{A}, C)$ with $L(\mathcal{A}, C) = L$.
(ii) L is definable in the existential fragment of WS1S$^{\text{card}}$*, that is, there is an existential* WS1S$^{\text{card}}$ *formula φ with $L(\varphi) = L$.*

Another extension of the classical logic-automata connection with a similar flavor is given in [22] relating Petri net languages with the existential fragment of the MSO logic on words extended with partial orders $\leq_g$ and $=_g$ on subsets of $\{0, \ldots, n-1\}$ defined as $X \leq_g Y$ iff $|X \cap \{0, \ldots, m-1\}| \leq |Y \cap \{0, \ldots, m-1\}|$, for all $m \leq n$, and $X =_g Y$ iff $X \leq_g Y$ and $|X| = |Y|$. [22] does not investigate decidability problems about this logic as we will do in the next section for WS1S$^{\text{card}}$.

We want to point out that there is also a relationship between WS1S$^{\text{card}}$ and Petri nets. Petri net reachability is expressible in WS1S$^{\text{card}}$ [14]. From this it is not difficult to see that ($\overline{0}$-closed) Petri net languages can be described in WS1S$^{\text{card}}$. But the formulas for expressing the reachability problem (or describing Petri net languages) require a top-level quantification of the form $\exists x \exists \overline{X} \forall y \forall \overline{Y}$. In the next section we are going to show that the fragment with such a top-level quantification is undecidable. Note that the reachability problem for Petri nets is decidable.

4 Undecidability and Decidability Results

Decidability and undecidability results about MSO logics with cardinality constraints summarized in Figure 1 using the notation introduced below. Together, these results delimit the boundary between decidability versus undecidability in WS1S$^{\text{card}}$. Furthermore, we illustrate applications in hardware and software verification of a decidable fragment of WS1S$^{\text{card}}$.

We introduce the following notation to uniformly describe fragments of WS1S$^{\text{card}}$.

undecidable	$[\forall_{\mathrm{MSO}}\exists^*_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$	$[\exists_{\mathrm{MSO}}\forall_{\mathrm{FO}}\exists^5_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$
decidable	$[\mathsf{FO}(\exists^*_{\mathrm{MSO}}\cup\forall^*_{\mathrm{MSO}})\mathsf{FO};\mathsf{R}_{\mathrm{WS1S}}]$	

Fig. 1. Undecidable and decidable fragments of $\mathrm{WS1S}^{\mathrm{card}}$.

Definition 13. *Let* $\mathsf{Q}\subseteq\{\exists_{\mathrm{MSO}},\forall_{\mathrm{MSO}},\exists_{\mathrm{FO}},\forall_{\mathrm{FO}}\}^*$ *and let* R *be a set of relations over the natural numbers and finite subsets of natural numbers. We write* $[\mathsf{Q};\mathsf{R}]$ *for the set of sentences of the form* $Q_1\alpha_1\dots Q_n\alpha_n\varphi$*, where* $Q_1\dots Q_n\in\mathsf{Q}$ *and* φ *is a quantifier-free formula with relations in* R*.*

We write $[\mathsf{Q};R_1,\dots,R_n]$ for $[\mathsf{Q};\{R_1,\dots,R_n\}]$. Let $\mathsf{R}_{\mathrm{WS1S}}$ be the set of relations that are definable in WS1S. We will often give the set Q as a regular expression. For example, the set FO of arbitrary FO quantifier prefixes is $(\exists_{\mathrm{FO}}\cup\forall_{\mathrm{FO}})^*$, and we write, e.g., $\exists^2_{\mathrm{MSO}}$ for $\exists_{\mathrm{MSO}}\exists_{\mathrm{MSO}}$.

Undecidability Results.

Theorem 14. *The fragment* $[\forall_{\mathrm{MSO}}\exists^*_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$ *is undecidable.*

Proof. To prove this theorem we look at the universality problem for $\overline{0}$-closed PFWAs which is undecidable. This can be shown by adopting the proof that shows the undecidability for universality problem for PFWAs.

Let $(\mathcal{A},C)$ be a $\overline{0}$-closed PFWA with $L(\mathcal{A},C)\subseteq\{0,1\}^*$. For the formula $\psi_{\mathcal{A},C}(U_1)$ from Theorem 10, it holds, $(\mathbb{N},\mathrm{succ})\models\forall U_1\psi_{\mathcal{A},C}$ iff $L(\mathcal{A},C)=\{0,1\}^*$. Since $\psi_{\mathcal{A},C}$ is existential and the universality problem is undecidable, we have that the fragment $[\forall_{\mathrm{MSO}}\exists^*_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$ is undecidable. ⊣

Theorem 15. *The fragment* $[\exists_{\mathrm{MSO}}\forall_{\mathrm{FO}}\exists^5_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$ *is undecidable.*

Proof (Sketch). The undecidability is shown by encoding the halting problem for 2-register machines as a formula in $[\exists_{\mathrm{MSO}}\forall_{\mathrm{FO}}\exists^5_{\mathrm{MSO}}\mathsf{FO};\mathrm{succ}]$.

Let $\mathcal{C}$ be a 2-register machine. A computation of $\mathcal{C}$ can be encoded as a word $w\in\{0,1\}^*$ in the following way. The word w consists of segments of the form $110b_1\dots b_s0z_0z_1z_0'z_1'$. The sequence $b_1\dots b_s$ encodes the state, namely $b_q=1$ iff the state of the configuration is q. The sequence $z_0z_1z_0'z_1'$ encodes the increment or decrement of a register: $z_i=1$ iff the ith register is incremented, and $z_i'=1$ iff the ith register is decremented. With the letters $110\dots0\dots$ we can check whether a subword of w represents an encoding of a configuration.

We define a sentence of the form $\exists X\forall y\exists U\exists Z_0\exists Z_1\exists Z_0'\exists Z_1'\psi$, where ψ is FO. The details on this sentence are in [15]. Intuitively, X represents a word $w\in\{0,1\}^*$ that is an encoding of a computation of $\mathcal{C}$, where w is a concatenation of sequences of the form $110\dots0\dots$ as explained above. The FO variable y intuitively ranges over all the configurations in X, and the MSO variables Z_i,Z_i' take care of the increments and decrements of the ith register up to the yth configuration. Therefore, $|Z_i|-|Z_i'|$ is the value of the ith counter in the yth configuration. The MSO variable U is used for some technical reasons; it represents the set $\{0,\dots,y\}$. ⊣

Decidability Result. Since the emptiness problem for PFWAs is decidable and the construction of the PFWA in Theorem 11 for a given formula in E is constructive, we get a decision procedure for E: the formula is satisfiable iff the language of the constructed PFWA is nonempty. Here we show a stronger decidability result. Namely, we give a decision procedure for sentences that have an arbitrary prefix of FO quantifiers and the body of the sentence or its negation is in E. This is done by two constructions. We first construct a PFWA using Theorem 11, where we drop the prefix of FO quantifiers of the given sentence. Second, we construct from this PFWA a formula in Presburger arithmetic taking care of the quantification of the FO variables.

Theorem 16. *The fragment* $[\mathsf{FO}(\exists^*_{\mathrm{MSO}} \cup \forall^*_{\mathrm{MSO}})\mathsf{FO}; \mathsf{R}_{\mathrm{WS1S}}]$ *is decidable.*

Proof. Case I: $\varphi \in [\mathsf{FO}\exists^*_{\mathrm{MSO}}\mathsf{FO}; \mathsf{R}_{\mathrm{WS1S}}]$. Note that every relation R occurring in φ is expressible by a WS1S formula ψ_R. Therefore, we can assume that φ is of the form $Q_1 x_1 \dots Q_m x_m \varphi'$ with $Q_1, \dots, Q_m \in \{\exists, \forall\}$ and $\varphi' \in \mathsf{E}$ by substituting the relations R with ψ_R. By Theorem 11 we can construct a PFWA $(\mathcal{A}, C)$ with dimension $N \geq 1$ and $L(\varphi') = L(\mathcal{A}, C)$. Assume that $\mathcal{A} = (S, \Gamma, \delta, s_{\mathrm{I}}, F)$ with $\Gamma \subseteq \{0,1\}^m \times \mathbb{N}^N$. It holds, $(\mathbb{N}, \mathrm{succ}) \models Q_1 x_1 \dots Q_m x_m \varphi'$ iff

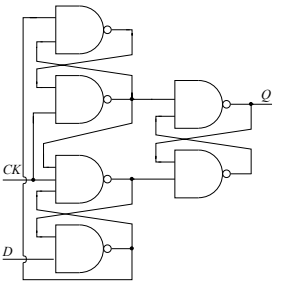
$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there is a word } w \in L(\mathcal{A}, C) \text{ such that} \\ &\quad I_w(x_1) = \tilde{x}_1, \dots, I_w(x_m) = \tilde{x}_m \,. \end{aligned} \tag{1}$$

By definition, (1) is equivalent to

$$\begin{aligned} &Q_1 \tilde{x}_1 \in \mathbb{N}, \dots, Q_m \tilde{x}_m \in \mathbb{N} \text{ there is a word } w \in L(\mathcal{A}) \text{ such that} \\ &\quad I_{\Psi(w)}(x_1) = \tilde{x}_1, \dots, I_{\Psi(w)}(x_m) = \tilde{x}_m \text{ and } \Phi(w) \in C \,, \end{aligned} \tag{2}$$

where $\Psi : \Gamma \to \{0,1\}^m$ is the projection and Φ the extended Parikh map of Γ.

We extend the alphabet Γ to $\Gamma' := \{(b, \overline{v}, \overline{v'}) \,|\, (b, \overline{v}) \in \Gamma \text{ and } \overline{v}' \in \{0,1\}^m\}$, that is, we append the vectors in $\{0,1\}^m$ to each symbol $(b, \overline{v}) \in \Gamma$. Let Φ' be the extended Parikh map of Γ', and let $h : \Gamma'^* \to \Gamma^*$ be the homomorphism defined by $h(b, \overline{v}, \overline{v}') := (b, \overline{v})$. We construct an automaton $\mathcal{A}'$ accepting $w \in \Gamma'^*$ iff $h(w) \in L(\mathcal{A})$ and $\Phi'(w) = \big(\Phi(h(w)), I_{\Psi(h(w))}(x_1), \dots, I_{\Psi(h(w))}(x_m)\big)$. 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 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 x_i, that is $B_i := \bigcup_{j \in \mathbb{N}} B_i^j$, where $B_i^0 := \{s_{\mathrm{I}}\}$ and for $j > 0$, $B_i^{j+1} := \{s \in \delta(B_i^j, (b, \overline{v})) \,|\, (b, \overline{v}) \in \Gamma \text{ with } \chi_{x_i}(b) = 0\}$. Note that if a state s is in B_i and from s we can still reach an accepting state then for every word $w \in \Gamma^*$ with $\delta(s_{\mathrm{I}}, w) = s$ it holds that $\chi_{x_i}(w)$ is of the form $0 \dots 0$. Otherwise, there would be a word in $L(\mathcal{A}, C)$ that is not an interpretation for the FO variable x_i. For $s \in S$, let $\overline{c}(s) \in \{0,1\}^m$ be the characteristic vector of s, that is $\overline{c}(s) := (c_1, \dots, c_m)$, where $c_i = 1$ iff $s \in B_i$. Now, $\delta' : S \times \Gamma' \to \mathcal{P}(S)$ is defined by $\delta'(s, (b, \overline{v}, \overline{v}')) := \{s' \in \delta(s, (b, \overline{v})) \,|\, \overline{c}(s') = \overline{v}'\}$. By the construction of $\mathcal{A}'$, (2) is equivalent to



If
- the clock CK has a rising edge at time t and the next rising edge of CK is at time t', and
- CK is stable from d_1 units of time after t and CK is stable d_2 units of time before t ($d_1 + d_2$ is the *minimum clock period*), and
- D is stable d_3 units of time up to time t (d_3 is the setup time),

then
- Q is stable from d_4 units of time after t (d_4 is the *start time*) until d_5 units of time after t' (d_5 is the *finish time*), and
- at time t', Q equals D at time t.

Fig. 2. Circuit of an edge-triggered D-type flip-flop and its specification.

$$Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there is a word } w \in L(\mathcal{A}') \text{ such that } \Phi(h(w)) \in C \text{ and } \Phi'(w) = (\Phi(h(w)), \tilde{x}_1, \ldots, \tilde{x}_m)\,. \tag{3}$$

From Lemma 5, we know that $\Phi'(L(\mathcal{A}'))$ is the union of the images of linear polynomials $q_1, \ldots, q_\ell : \mathbb{N}^r \to \mathbb{N}^N$, for some $r \geq 1$. Moreover, these polynomials are constructible from $\mathcal{A}'$. We conclude that (3) is equivalent to

$$Q_1\tilde{x}_1 \in \mathbb{N}, \ldots, Q_m\tilde{x}_m \in \mathbb{N} \text{ there are } y_1, \ldots, y_r \in \mathbb{N} \text{ and } \overline{v} \in \mathbb{N}^N \text{ such that } \overline{v} \in C \text{ and } q_i(y_1, \ldots, y_r) = (\overline{v}, \tilde{x}_1, \ldots, \tilde{x}_m), \text{ for some } 1 \leq i \leq \ell\,. \tag{4}$$

Note that (4) can be expressed as a sentence in Presburger arithmetic. The claim follows from the decidability of Presburger arithmetic.

Case II: $\varphi \in [\mathsf{FO}\forall^*_{\mathrm{MSO}}\mathsf{FO}; \mathsf{R}_{\mathrm{WS1S}}]$. Follows from Case I by the duality of quantifiers. ⊣

Applications. As an application, we sketch how this decidable fragment can be used to decide WS1S extended with some restricted linear arithmetic. Our example is the verification of an edge-triggered D-type flip-flop, taken from [3,8]. Although the circuit is built from only six nand-gates (left half of Figure 2), proving that the circuit meets its specification (right half of Figure 2) is "fairly complicated", as Gordon noted in [8]. The proof in [8] was done by paper and pencil, and contained a flaw, as reported in [3,28]. The correctness proof in [3] was done automatically by naturally expressing the higher-order logic formalization from [8] in WS1S and using the implementation of the automata-based decision procedure for WS1S in the Mona tool [10]. This verification technique works only if the parameters $d_1, \ldots, d_5$ are instantiated with concrete values because the specification contains some linear arithmetic, for example, "Q is stable from d_4 units of time after t until d_5 units of time after t'". Reusing most of the WS1S formalization from [3] we can formalize in the decidable fragment of WS1S$^{\mathrm{card}}$ whether the circuit meets its specification for all $d_1, \ldots, d_5 \in \mathbb{N}$ satisfying, for instance, the constraints $d_1 \geq 2$, $d_2 \geq 2$, $d_1 + d_2 \geq 5$, $d_3 \geq 3$, $d_4 \geq 3$, and $d_5 \leq 2$. Together with Theorem 16 this demonstrates that such parameterized verification problems are actually decidable problems.

We briefly recall the formalization in [3].[1] To keep the formulas readable, we use some syntactic sugar for WS1S. It will always be straightforward to translate the used notation to WS1S. Note that $x \leq y$ can be defined by $\forall Z(Zy \wedge \forall z(Zz + 1 \rightarrow Zz) \rightarrow Zx)$.

The temporal behavior of a unit-delay nand-gate with inputs X and Y, and output Z up to time \$ is described by $nand(\$, X, Y, Z) := \forall t\big(t < \$ \rightarrow \big(Zt+1 \leftrightarrow \neg(Xt \wedge Yt)\big)\big)$, where \$ is an FO variable and X, Y, and Z are MSO variables. The temporal behavior of a nand-gate with three inputs can be described analogously. The circuit of the left half of Figure 2 implementing a D-type flip-flop can now be described by the following formula, where the internal wires are hidden by existential quantification.

$$\begin{aligned} imp(\$, D, CK, Q) := \exists W_1 \exists W_2 \exists W_3 \exists W_4 \exists W_5 \big(&nand(\$, W_2, D, W_1) \wedge \\ &nand3(\$, W_3, CK, W_1, W_2) \wedge nand(\$, W_4, CK, W_3) \wedge \\ &nand(\$, W_1, W_3, W_4) \wedge nand(\$, W_3, W_5, Q) \wedge \\ &nand(\$, Q, W_2, W_5)\big) \end{aligned}$$

We recall the definitions [3] of the temporal concepts needed to formalize the flip-flop's specification.

- X is stable in the interval $[t, t')$:

$$stable(t, t', X) := \forall u\big(t \leq u < t' \rightarrow (Xu \leftrightarrow Xt)\big)$$

- X rises at t:

$$rise(t, X) := t > 0 \wedge \neg X(t-1) \wedge Xt$$

- t' is the next instance after t where X rises:

$$nextRise(t, t', X) := rise(t', X) \wedge \forall u\big(t < u < t' \rightarrow \neg rise(u, X)\big)$$

The flip-flop's specification given in the right half of Figure 2 can be formalized as

$$\begin{aligned} spec(\$, t, t', D, CK, Q) := \big(&d_2 \leq t < t' \leq \$ - d_5 \wedge \\ &rise(t, CK) \wedge nextRise(t, t', CK) \wedge \\ &stable(t, t + d_1, CK) \wedge stable(t - d_2, t, CK) \wedge \\ &stable(t + 1 - d_3, t + 1, D)\big) \rightarrow \\ &stable(t + d_4, t' + d_5, Q) \wedge (Qt' \leftrightarrow Dt)\,. \end{aligned}$$

Note that this formula is a WS1S formula if $d_1, \ldots, d_5$ are not FO variables but natural numbers. For fixed values for $d_1, \ldots, d_5$, Mona checks automatically if the circuit meets its specification by computing the truth value of the formula

$$verify(\$, t, t', D, CK, Q) := imp(\$, D, CK, Q) \rightarrow spec(\$, t, t', D, CK, Q)\,.$$

[1] Actually, Basin and Klarlund did not use WS1S but M2L(str). There are some technical differences between these logics as explained in §3. We have adopted their formalization to WS1S.

In the following, we show how the decidability result from Theorem 16 can be used to check whether the circuit is correct, for instance, for all $d_1, \ldots, d_5 \in \mathbb{N}$ with $d_1 \geq 2$, $d_2 \geq 2$, $d_1 + d_2 \geq 5$, $d_3 \geq 3$, $d_4 \geq 3$, and $d_5 \leq 2$. The constraints on the parameters can be expressed in WS1S by

$$\mathit{constr}(d_1, \ldots, d_5) := (d_1 \geq 2 \wedge d_2 \geq 3 \vee d_1 \geq 3 \wedge d_2 \geq 2) \wedge d_3 \geq 3 \wedge d_4 \geq 3 \wedge d_5 \leq 2\,.$$

Unfortunately, $\forall d_1 \ldots \forall d_5 \big(\mathit{constr}(d_1, \ldots, d_5) \rightarrow \forall \$ \forall t \forall t' \forall D \forall CK \forall Q \ \big(\mathit{verify}(\$, t, t', D, CK, Q)\big)\big)$ is not a $\mathrm{WS1S}^{\mathrm{card}}$ formula, since *verify* contains in the subformula *spec* terms involving linear arithmetic, for example, $t + d_1$. But we can take a detour using MSO variables. For example, for the term $t + d_1$, we introduce an FO variable x_{t+d_1} and MSO variables T, D_1 with $x_{t+d_1} = |T| + |D_1|$, $T = \{0, \ldots, t-1\}$, and $D_1 = \{0, \ldots, d_1 - 1\}$. It holds $x_{t+d_1} = t + d_1$. Thus, the term $t + d_1$ can be substituted by x_{t+d_1}. Let *spec*′ be the formula, where we replace in *spec* the terms τ involving linear arithmetic by fresh variables x_τ, that is,

$$\begin{aligned}
&\mathit{spec}'(\$, t, t', D, CK, Q, x_{\$-d_5}, x_{t+d_1}, x_{t-d_2}, x_{t-d_3}, x_{t+d_4}, x_{t'+d_5}) := \\
&\qquad \big(d_2 \leq t < t' \leq x_{\$-d_5} \wedge \\
&\qquad\ \ \mathit{rise}(t, CK) \wedge \mathit{nextRise}(t, t', CK) \wedge \\
&\qquad\ \ \mathit{stable}(t, x_{t+d_1}, CK) \wedge \mathit{stable}(x_{t-d_2}, t, CK) \wedge \\
&\qquad\ \ \mathit{stable}(x_{t-d_3} + 1, t + 1, D)\big) \ \rightarrow \\
&\qquad \mathit{stable}(x_{t+d_4}, x_{t'+d_5}, Q) \wedge (Qt' \leftrightarrow Dt)\,.
\end{aligned}$$

We write $x = \pm|X_1| \pm \cdots \pm |X_r|$ for $\exists Z \big(\forall z (Zz \leftrightarrow z < x) \wedge |Z| = \pm|X_1| \pm \cdots \pm |X_r|\big)$, where x is an FO variable and the X_is are MSO variables. The formula *aux* ensures that the new variables in *spec*′ have the correct values, for example, x_{t+d_1} equals $t + d_1$.

$$\begin{aligned}
&\mathit{aux}(d_1, \ldots, d_5, \$, t, t', x_{\$-d_5}, x_{t+d_1}, x_{t-d_2}, x_{t-d_3}, x_{t+d_4}, x_{t'+d_5}) := \\
&\qquad \exists D_1 \ldots \exists D_5 \exists \pounds \exists T \exists T' \big(d_1 = |D_1| \wedge \cdots \wedge d_5 = |D_5| \wedge \\
&\qquad\qquad\qquad \$ = |\pounds| \wedge t = |T| \wedge t' = |T'| \wedge \\
&\qquad\qquad\qquad x_{\$-d_5} = |\pounds| - |D_5| \wedge x_{t+d_1} = |T| + |D_1| \wedge \\
&\qquad\qquad\qquad x_{t-d_2} = |T| - |D_2| \wedge x_{t-d_3} = |T| - |D_3| \wedge \\
&\qquad\qquad\qquad x_{t+d_4} = |T| + |D_4| \wedge x_{t'+d_5} = |T'| + |D_5|\big)
\end{aligned}$$

For proving the circuit correct, we have to check whether the formula

$$\mathit{verify}' := \mathit{aux} \wedge \mathit{constr} \rightarrow (\mathit{imp} \rightarrow \mathit{spec}')$$

is valid. This can be done automatically by Theorem 16, since *verify*′ can be transformed into a formula in $[\mathsf{FO}\forall^*_{\mathrm{MSO}}\mathsf{FO}; \mathsf{R}_{\mathrm{WS1S}}]$ by universally quantifying over the FO variables and the MSO variables D, CK, Q, and by pulling out the existentially quantified MSO variables in *aux*. Note that the existential quantifiers become universal by this process. In addition to verification, our procedure may also be used for synthesizing sufficient parameter constraints if we do not

restrict the parameters $d_1, \ldots, d_5$ by some constraints and do not universally quantify over them.

Although our decision procedure is built on top of a decision procedure for Presburger arithmetic and a translation from $\text{WS1S}^{\text{card}}$ formulas to PFWAs and the worst-case complexity is in both cases very high we are encouraged by the outcomes with a prototype implementation. We tested our implementation on various case studies, such as the D-type flip-flop above and lemmas in a PVS theory about cardinalities of finite sets that were used in [24] to verify oral message algorithms. Such kinds of proofs are cumbersome and rather involved. Our decidability result opens up the possibility to effectively automate such kinds of verification problems.

5 Conclusions

We have extended WS1S with linear cardinality constraints, proved the undecidability of this extension, and identified decidable fragments (see Figure 1). These results were obtained by extending the logic-automata connection to fragments of WS1S with cardinality constraints and an appropriate automaton model that we call *Parikh finite word automata*. The resulting decision procedure has applications in both hardware and protocol verification [14, 15], and initial experiments with an extension of the Mona tool with cardinality constraints are encouraging [13].

One advantage of our notion of Parikh word automata is that it easily generalizes to trees. A decidability result for a fragment of the weak monadic second-order logic of *two* successors with cardinality constraints using *Parikh finite tree automata* is included in [15]. Since monadic second-order logics on trees give a theoretical foundation of XML query languages [9], our results on trees may serve as a theoretical basis for extending current query languages as in [5].

The framework in §2 can also be generalized to infinite words and trees. A possible acceptance condition is in the spirit of the Büchi acceptance condition: one requires that the arithmetic constraints have to be satisfied for infinitely many prefixes in order to accept the input. Another extension that we want to look at is to generalize the framework to graphs with bounded tree-width.

Future work will include detailed complexity analyses, theoretically and practically, on Parikh automata and on the decision procedure for the decidable fragment of $\text{WS1S}^{\text{card}}$.

Acknowledgments. We thank J. Rushby for initiating and supporting this research, and the anonymous referees for their invaluable comments. The first author also thanks J. Meseguer.

References

1. D. Basin and S. Friedrich, *Combining WS1S and HOL*, in FroCos'98, Applied Logic Series, 2000, pp. 39–56.

2. D. Basin, S. Friedrich, and S. Mödersheim, *B2M: A semantic based tool for BLIF hardware descriptions*, in FMCAD'00, vol. 1954 of LNCS, 2000, pp. 91–107.
3. D. Basin and N. Klarlund, *Automata based symbolic reasoning in hardware verification*, FMSD, 13 (1998), pp. 255–288.
4. H. Comon and Y. Jurski, *Multiple counters automata, safety analysis and Presburger arithmetic*, in CAV'98, vol. 1427 of LNCS, 1998, pp. 268–279.
5. S. Dal Zilio and D. Lugiez, *XML schema, tree logic and sheaves automata*, Research Report 4631, INRIA, 2002.
6. J. Dassow and V. Mitrana, *Finite automata over free groups*, International Journal of Algebra and Computation, 10 (2000), pp. 725–737.
7. 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.
8. M. Gordon, *Why higher-order logics is a good formalism for specifying and verifying hardware*, in Formal Aspects of VLSI Design, North-Holland, 1986, pp. 153–177.
9. G. Gottlob and C. Koch, *Monadic Datalog and the expressive power of languages for web information extraction*, in PODS'02, 2002, pp. 17–28.
10. 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.
11. O. Ibarra, *Reversal-bounded multicounter machines and their decision problems*, JACM, 25 (1978), pp. 116–133.
12. O. Ibarra, J. Su, Z. Dang, T. Bultan, and R. Kemmerer, *Counter machines and verification problems*, TCS, 289 (2002), pp. 165–189.
13. F. Klaedtke, *CMona: Monadic second-order logics with linear cardinality constraints in practice.* in preparation, 2003.
14. F. Klaedtke and H. Ruess, *WS1S with cardinality constraints*, Technical Report SRI-CSL-05-01, SRI International, 2001.
15. ———, *Parikh automata and monadic second-order logics with linear cardinality constraints*, Technical Report 177, Albert-Ludwigs-Universität Freiburg, 2002. (revised version).
16. N. Klarlund, A. Møller, and M. Schwartzbach, *MONA implementation secrets*, in CIAA'00, vol. 2088 of LNCS, 2000, pp. 182–194.
17. N. Klarlund, M. Nielsen, and K. Sunesen, *Automated logical verification based on trace abstraction*, in PODC'96, 1996, pp. 101–110.
18. L. Lamport, R. Shostak, and M. Pease, *The Byzantine Generals problem*, TOPLAS, 4 (1982), pp. 382–401.
19. A. Meyer, *Weak monadic second-order theory of successor is not elementary-recursive*, in Logic Colloquium, vol. 453 of LNM, 1975, pp. 132–154.
20. V. Mitrana and R. Stiebe, *Extended finite automata over groups*, Discrete Applied Mathematics, 108 (2001), pp. 287–300.
21. S. Owre and H. Ruess, *Integrating WS1S with PVS*, in CAV'00, vol. 1855 of LNCS, 2000, pp. 548–551.
22. M. Parigot and E. Pelz, *A logical approach of Petri net languages*, TCS, 39 (1985), pp. 155–169.
23. R. Parikh, *On context-free languages*, JACM, 13 (1966), pp. 570–581.
24. J. Rushby, *Systematic formal verification for fault-tolerant time-triggered algorithms*, IEEE Trans. on Software Engineering, 2 (1999), pp. 651–660.
25. M. Smith and N. Klarlund, *Verification of a sliding window protocol using IOA and MONA*, in FORTE/PSTV'00, vol. 183 of IFIP Conf. Proc., 2000, pp. 19–34.
26. L. Stockmeyer, *The Complexity of Decision Problems in Automata Theory and Logic*, PhD thesis, Dept. of Electrical Engineering, MIT, Boston, Mass., 1974.

27. W. Thomas, *Languages, automata, and logic*, in Handbook of Formal Languages, vol. 3, Springer-Verlag, 1997, pp. 389–455.
28. A. Wilk and A. Pnueli, *Specification and verification of VLSI systems*, in ICCAD'89, 1989, pp. 460–463.