# Semënov Arithmetic, Affine VASS, and String Constraints

**Andrei Draghici**
Department of Computer Science, University of Oxford, UK

**Christoph Haase**
Department of Computer Science, University of Oxford, UK

**Florin Manea**
Computer Science Department and Campus-Institut Data Science, Göttingen University, Germany

## Abstract

We study extensions of Semënov arithmetic, the first-order theory of the structure $\langle \mathbb{N}, +, 2^x \rangle$. It is well-known that this theory becomes undecidable when extended with regular predicates over tuples of number strings, such as the Büchi $V_2$-predicate. We therefore restrict ourselves to the existential theory of Semënov arithmetic and show that this theory is decidable in EXPSPACE when extended with arbitrary regular predicates over tuples of number strings. Our approach relies on a reduction to the language emptiness problem for a restricted class of affine vector addition systems with states, which we show decidable in EXPSPACE. As an application of our result, we settle an open problem from the literature and show decidability of a class of string constraints involving length constraints.

**2012 ACM Subject Classification** Theory of computation → Logic

**Keywords and phrases** arithmetic theories, Büchi arithmetic, exponentiation, vector addition systems with states, string constraints

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2024.29

**Funding** This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 852769, ARiAT). Florin Manea's work was supported by a Heisenberg Grant, funded by the Deutsche Forschungsgemeinschaft (DFG, project 466789228).

## 1 Introduction

This paper studies the decidability and complexity of the existential theory of an extension of the structure $\langle \mathbb{N}, 0, 1, +, 2^x \rangle$, where $2^x$ is the function mapping a natural number $n$ to $2^n$. Decidability of the first-order theory of this structure was first shown by Semënov in a more general framework using an automata-theoretic approach [13], and we henceforth call this theory *Semënov arithmetic*. As shown by Cherlin and Point [7], see also [11], Semënov arithmetic admits quantifier elimination and has a quantifier-elimination procedure that runs in non-elementary time, and this upper bound is tight [11]. The existential fragment of Semënov arithmetic has recently been shown decidable in NEXP [2] by giving a more elaborate quantifier elimination procedure. Unlike its substructure Presburger arithmetic (obtained by dropping the function $2^x$), Semënov arithmetic is not automatic in the sense of the theory of automatic structures [5, 10]. Consider[1] the family of formulas $\Phi_n \equiv x_1 = 1 \wedge \bigwedge_{1 \le i \le n} x_{i+1} = 2^{x_i} \wedge y < x_n$. Then the (finite) number of solutions of $\Phi_n$ is a tower of height $n$. Suppose Semënov arithmetic (viewed as a relational structure) was automatic. Then each of its relations is definable by a deterministic finite automaton (DFA)

[1] We thank an anonymous reviewer for suggesting this argument.

of size at most $m$ for some $m \in \mathbb{N}$ over some alphabet $\Sigma$. But then the DFA for $\Phi_n$ is acyclic and has at most $m^{n+2}$ many states, meaning that $\Phi_n$ has at most $(m^{(n+2)}|\Sigma|)^{(m^{n+2})}$ different solutions, a contradiction.

The decidability of Semënov arithmetic is fragile with respect to extensions of the structure. For instance, it is not difficult to see that extending Semënov arithmetic with the Büchi predicate $V_2(x, y)$, where $V_2(x, y)$ holds whenever $x$ is the largest power of two dividing $y$ without remainder, results in an undecidable first-order theory, see e.g. [11]. However, this undecidability result requires an $\exists^*\forall^*$-quantifier prefix and does not rule out decidability of the existential fragment. The main result of this paper is to show that the existential theory of *generalised Semënov arithmetic*, i.e., the existential theory of $\langle \mathbb{N}, 0, 1, +, 2^x, \{R_i\}_{i\geq 0}\rangle$, where $R_0, R_1, \ldots$ is an enumeration of all regular languages over the alphabets $\{0,1\}^d$, $d \geq 1$, is decidable in EXPSPACE. Non-automaticity of Semënov arithmetic and undecidability of $\langle \mathbb{N}, 0, 1, +, 2^x, V_2\rangle$ rule out the possibility of approaching this existential theory via automatic structures based on finite-state automata or via quantifier-elimination *à la* Cherlin and Point, since $V_2$ is definable as a regular language over pairs of number strings. Instead, our decidability result is based on a reduction to the language non-emptiness problem of a special class of *affine vector addition systems with states (affine VASS).*

A VASS comprises a finite-state controller with a finite number of counters ranging over the natural numbers. In affine VASS, counters can be updated by affine functions $x \mapsto ax + b$ when taking a transition, provided that the resulting counter is non-negative. While reachability in affine VASS is decidable for a single counter [8], already in the presence of two counters reachability becomes undecidable [12]. Our reduction consequently requires a restricted class of affine VASS to obtain decidability. We call this class *restricted labelled affine VASS (restricted LAVASS).* A restricted LAVASS is an affine VASS with $d$ pairs of counters and hence $2d$ counters in total. For every pair, once the first counter achieves a non-zero value along a run, it keeps getting incremented at every transition; the second counter is only updated via affine functions $x \mapsto 2x$ and $x \mapsto 2x + 1$. A configuration consisting of a control state and $2d$ counter values is accepting whenever the control state is accepting and for every pair of counters, the first counter has the same value as the second counter. We give an EXPSPACE procedure for deciding emptiness of restricted LAVASS whose correctness proof is based on a counter elimination procedure in which we successively encode counters into a finite state space while preserving equi-non-emptiness. The EXPSPACE upper bound for existential generalised Semënov arithmetic follows from a reduction to language non-emptiness of a restricted LAVASS whose language encodes all solutions of the given formula. Note that obtaining an elementary upper bound is non-trivial since, e.g., the family of formulas $\Phi_n$ above shows that smallest solution of an existential formula can be non-elementary in bit-length.

As an application of our EXPSPACE upper bound for existential generalised Semënov arithmetic, we show that a certain class of string constraints with length constraints is decidable in EXPSPACE. This class allows for existentially quantifying over bit-strings, and to assert that the value of a string variable lies in a regular language, as well as Presburger-definable constraints over the lengths of the bit-strings stored in string variables and the numerical values of those variables (when viewed as encoding a number in binary). Decidability of this class was left open in [3]. We settle this open problem by showing that it can be reduced to the existential fragment of generalised Semënov arithmetic. Formulas of this class of string constraints appear widely in practice – in fact, all formulas in the extensive collection of standard real-world benchmark sets featured in [3, 4] lie in this class or can be reduced to formulas of this class by standard methods.

## 2 Preliminaries

### 2.1 Basic notation

By $\mathbb{Z}$ and $\mathbb{N}$ we denote the integers and non-negative integers, respectively. Given an $m \times n$ integer matrix $A$, we denote by $\|A\|_{1,\infty}$ the $(1,\infty)$-norm of $A$, which is the maximum over the sum of the absolute values of the elements of the rows in $A$. For $\boldsymbol{b} \in \mathbb{Z}^m$, $\|\boldsymbol{b}\|_\infty$ is the largest absolute value of the numbers occurring in $\boldsymbol{b}$.

### 2.2 Numbers as strings and strings as numbers

Here and below, let $\Sigma = \{0,1\}$ be a binary alphabet. Any string from $\Sigma^*$ has an interpretation as a binary encoding of a natural number, possibly with an arbitrary number of leading zeros. Conversely, any natural number in $\mathbb{N}$ can be converted into its bit representation as a string in $\Sigma^*$. Finally, by considering strings over $(\Sigma^k)^*$ for $k \geq 1$, we can represent $k$-tuples of natural numbers as strings over $\Sigma^k$, and *vice versa*. Formally, given $u = \boldsymbol{u}_n \boldsymbol{u}_{n-1} \ldots \boldsymbol{u}_0 \in (\Sigma^k)^*$, we define the tuple of natural numbers corresponding to $u$ in *most-significant digit first (msd)* notation as

$$[\![u]\!] := \sum_{i=0}^{n} 2^i \cdot \boldsymbol{u}_i \,.$$

Note that $[\![\cdot]\!]$ is surjective but not injective. We lift the definition of $[\![\cdot]\!]$ to sets in the natural way.

### 2.3 Generalised Semënov arithmetic

For technical convenience, the structures we consider in this paper are relational. We refer to the first-order theory of $\langle \mathbb{N}, 0, 1, +, 2^{(\cdot)} \rangle$ as *Semënov arithmetic*, where $+$ is the natural ternary addition relation, and $2^{(\cdot)}$ is the power relation of base two, consisting of all tuples $(a,b) \in \mathbb{N}^2$ such that $b = 2^a$. Semënov arithmetic is an extension of Presburger arithmetic, which is the first-order theory of the structure $\langle \mathbb{N}, 0, 1, + \rangle$. It is known that Semënov arithmetic is decidable and admits quantifier elimination [2, 7, 13].

For presentational convenience, atomic formulas of Semënov arithmetic are one of the following:

- linear equations of the form $a_1 \cdot x_1 + \cdots + a_d \cdot x_d = b$, $a_i, b \in \mathbb{Z}$, and
- exponential equations of the form $x = 2^y$.

Here, $x_1, \ldots, x_d, y$ are arbitrary first-order variables. Clearly, richer atomic formulas such as $x + 2^{2^y} + y = z + 5$ can be defined from those basic class of atomic formulas, since, in this example, $x + 2^{2^y} + y = z + 5 \equiv \exists u \exists v\, u = 2^v \wedge v = 2^y \wedge x + u + y - z = 5$. Moreover, since we are interpreting numbers over non-negative integers, we can define the order relation in existential Semënov arithmetic. This enables us to without loss of generality assume that existential formulas of Semënov arithmetic are positive, since $\neg(x = y) \equiv x < y \vee y < x$ and $\neg(x = 2^y) \equiv \exists z\, z = 2^y \wedge \neg(x = z)$.

The main contribution of this paper is to show that the existential fragment of a generalisation of Semënov arithmetic is decidable. Subsequently, we write $\mathbf{0}$ to denote a tuple of zeros in any arbitrary but fixed dimension. *Generalised Semënov arithmetic* additionally allows for non-negated atomic formulas $R(x_1, \ldots, x_k)$, where $R = \mathbf{0}^* \cdot L$ for some regular language $L \subseteq (\Sigma^k)^*$. We interpret $R$ as $[\![R]\!] \subseteq \mathbb{N}^k$, and the additional leading zeros we require ensure that $R = [\![ [\![R]\!] ]\!]^{-1}$. Subsequently, we call a language $L \subseteq (\Sigma^k)^*$ *zero closed* if $L = \mathbf{0}^* \cdot L$. Given a formula $\Phi(x_1, \dots, x_n)$ of generalised Semënov arithmetic, we define $[\![\Phi]\!] \subseteq \mathbb{N}^d$ as the set of all satisfying assignments of $\Phi$.

The size of an atomic formula $R(x_1, \dots, x_k)$ is defined as the number of states of the canonical minimal DFA defining $R$. For all other atomic formulas $\varphi$, we define their sizes $|\varphi|$ as the number of symbols required to write down $\varphi$, assuming binary encoding of numbers. The size $|\Phi|$ of an arbitrary existential formula $\Phi$ of generalised Semënov arithmetic is the sum of the sizes of all atomic formulas of $\Phi$.

The full first-order theory of generalised Semënov arithmetic is known to be undecidable [11]. This follows from the undecidability of $\langle \mathbb{N}, 0, 1, +, 2^{(\cdot)}, V_2 \rangle$, where $V_2$ is the binary predicate such that $V_2(x, y)$ holds if and only if $x$ is the largest power of 2 dividing $y$ without remainder. Note that $V_2$ can be defined in terms of a regular language, cf. [6]. The central result of this paper is the following:

▶ **Theorem 1.** *The existential fragment of generalised Semënov arithmetic is decidable in EXPSPACE.*

## 2.4 Affine vector addition systems with states

A technical tool for our decidability results is a tailor-made class of *labelled affine vector addition systems with states (LAVASS).* Formally, an LAVASS is a tuple $V = \langle Q, d, \Sigma, \Delta, \lambda, q_0, F, \Phi \rangle$, where $Q$ is a finite set of *control states*, $d \geq 0$ is the *dimension of* $V$, $\Sigma$ is a *finite alphabet*, $\Delta \subseteq Q \times \mathcal{P}(\Sigma) \times Q$ is a finite set of *transitions*, $\lambda \colon \Delta \to \mathit{Ops}^d$ is the *update function*, where $\mathit{Ops} \subseteq \mathbb{Z}[x]$ is the set of all affine functions over a single variable, $q_0 \in Q$ is the *initial control state*, $F \subseteq Q$ is the set of *final control states*, and $\Phi$ is a a quantifier-free formula of Presburger arithmetic $\Phi(x_1, \dots, x_d)$ that specifies a infinite set $[\![\Phi]\!] \subseteq \mathbb{N}^d$ of *final counter values.* Note that when $d = 0$ then $V$ is essentially a non-deterministic finite automaton.

The set of *configurations* of $V$ is $C(V) := Q \times \mathbb{N}^d$. The *initial configuration* of $V$ is $c_0 = (q_0, 0, \dots, 0)$, and the set of *final configurations* is $C_f(V) := \{(q_f, \boldsymbol{v}) : q_f \in F, \boldsymbol{v} \in [\![\Phi]\!]\}$ . For an update function $\lambda \colon \Delta \to \mathit{Ops}^d$, we define

$$\|\lambda\| := \max\{|a| + |b| : \lambda(t) = (f_1, \dots, f_d), f_i = ax + b, 1 \leq i \leq d, t \in \Delta\} .$$

We define the size $|V|$ of an LAVASS $V = \langle Q, d, \Sigma, \Delta, \lambda, q_0, F, S_f \rangle$ as

$$|V| := |Q| + |\Delta| \cdot (d + 1) \cdot \log(\|\lambda\| + 1) + |\Phi| .$$

An LAVASS induces an (infinite) labelled directed *configuration graph* $G = (C(V), \to)$, where $\to \subseteq C(V) \times \Sigma \times C(V)$ such that $c \xrightarrow{a} c'$ if and only if $c = (q, m_1, \dots, m_d)$ and $c' = (q', m_1', \dots, m_d')$ and there is $t = (q, A, q') \in \Delta$ such that $a \in A$, $\lambda(t) = (f_1, \dots, f_d)$, and $m_i' = f_i(m_i)$ for all $1 \leq i \leq d$. We lift the definition of $\to$ to words $w = a_1 \cdots a_n \in \Sigma^*$ in the natural way, and thus write $c \xrightarrow{w} c'$ whenever $c \xrightarrow{a_1} c_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} c_{n-1} \xrightarrow{a_n} c'$ for some $c_1, \dots, c_{n-1} \in C$. The *language* $L(V) \subseteq \Sigma^*$ of $V$ is defined as

$$L(V) := \{w \in \Sigma^* : c_0 \xrightarrow{w} c_f, c_f \in C_f(V)\} .$$

If we are interested in runs of LAVASS, we write $\pi = c_1 \xrightarrow{t_1} c_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-2}} c_{n-1} \xrightarrow{t_{n-1}} c_n$ to emphasise the sequence of configurations and transitions taken. For $1 \leq i \leq j \leq n$, we denote by $\pi[i, j]$ the subsequence $c_i \xrightarrow{t_i} c_{i+1} \xrightarrow{t_{i+1}} \cdots \xrightarrow{t_{j-1}} c_j$. We denote by $\mathit{val}(\pi, x_i)$ the value $m_i$ of the $i$-th counter in the last configuration of $\pi$.

The *non-emptiness problem* for an LAVASS is to decide whether $L(V) \neq \emptyset$. Affine VASS are a powerful class of infinite state systems, and even in the presence of only two counters and $\Phi(x_1, x_2) \equiv x_1 = x_2$, the emptiness problem is undecidable [12]. In Section 4, we identify a syntactic fragment of LAVASS for which non-emptiness can be decided in EXPSPACE.

We briefly discuss closure properties of LAVASS and show that they are closed under union and intersection, and restricted kinds of homomorphisms and inverse homomorphisms, using essentially the standard constructions known for finite-state automata. Let $V_i = \langle Q_i, d_i, \Sigma, \Delta_i, \lambda_i, q_0^{(i)}, F_i, \Phi_i\rangle$, $i \in \{1, 2\}$, be two LAVASS.

▶ **Proposition 2.** *The languages of LAVASS are closed under union and intersection. Moreover, for $V$ such that $L(V) = L(V_1) \cap L(V_2)$, we have $|V| \leq |V_1| \cdot |V_2|$.*

**Proof.** This result can be obtained by generalising the standard constructions known from non-deterministic finite-state automata. The set of control states of the LAVASS $V$ accepting the intersection of LAVASS $V_1$ and $V_2$ is $Q_1 \times Q_2$. The dimension of $V$ is the sum of the dimensions of $V_1$ and $V_2$, and the counters of $V_1$ and $V_2$ get independently simulated in the counters of $V$. Upon reading an alphabet symbol $a$, the LAVASS $V$ then simultaneously simulates the respective transitions of $V_1$ and $V_2$ for $a$; further details are relegated to Appendix A.1. ◀

Note that since LAVASS languages contain regular languages, Proposition 2 in particular enables us to intersect LAVASS languages with regular languages.

Let $\Sigma, \Gamma$ be two finite alphabets. Recall that a homomorphism $h\colon \Gamma^* \to \Sigma^*$ is fully defined by specifying $h(a)$ for all $a \in \Gamma$. We call $h$ a *projection* if $|h(a)| = 1$ for all $a \in \Gamma$.

▶ **Proposition 3.** *The languages of LAVASS are closed under projections and inverses of projections.*

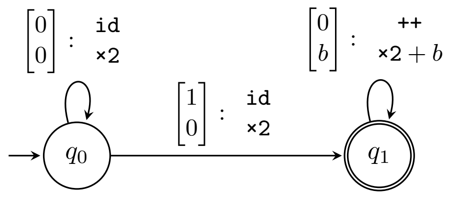
**Proof.** Let $h\colon \Gamma^* \to \Sigma^*$ be a projection. Given an LAVASS $V = \langle Q, d, \Sigma, \Delta, \lambda, q_0, F, S_f\rangle$, to obtain closure under projections replace any $t = (q, A, q') \in \Delta$ with $t' = (q, h(A), q')$, and set $\lambda(t') := \lambda(t)$. To obtain closure under inverse projections, replace any $t = (q, A, q') \in \Delta$ with $t' = (q, h^{-1}(A), q')$ and set $\lambda(t') := \lambda(t)$. ◀

## 3 Reducing Semënov arithmetic to restricted LAVASS

Let $\Sigma = \{0, 1\}$. In this section, we show how given a quantifier-free formula $\Phi(x_1, \ldots, x_d)$ of Semënov arithmetic, we can construct an LAVASS $V$ over the alphabet $\Sigma_d := \{0, 1\}^d$ such that $[\![L(V)]\!] = \{\boldsymbol{x} \in \mathbb{N}^d : \Phi(\boldsymbol{x})\}$. We will subsequently observe that the resulting LAVASS enjoy strong structural restrictions, giving rise to the fragment of restricted LAVASS that we then formally define. For our purposes, it will be sufficient to primarily focus on formulas $\Phi$ of Semënov arithmetic which are conjunctions of atomic formulas.

In the previous section, we stated that for presentational convenience, the atomic formulas of Semënov arithmetic are either linear equations or exponential equations of the type $x = 2^y$. It is well known that the sets of solutions of systems of linear equations $A \cdot \boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{Z}$ can be represented by a regular language and are hence definable via an LAVASS.

▶ **Lemma 4** ([14], see also [9, Eqn. (1)]). *Given a system of equations $\Phi \equiv A \cdot \boldsymbol{x} = \boldsymbol{b}$ with $A \in \mathbb{Z}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{Z}^m$, there is a DFA $V$ with at most $2^m \cdot \max\{\|A\|_{1,\infty}, \|\boldsymbol{b}\|_\infty\}^m$ states such that $L(V)$ is zero-closed and $[\![L(V)]\!] = [\![\Phi]\!]$.*



$\begin{bmatrix}0\\0\end{bmatrix}$ : id ×2
$\begin{bmatrix}1\\0\end{bmatrix}$ : id ×2
$\begin{bmatrix}0\\b\end{bmatrix}$ : ++ $\times 2 + b$
$q_0$
$q_1$

**Figure 1** A gadget with two counters for the exponential equation $x = 2^y$, where $b \in \{0, 1\}$.

The crucial part, which requires the power of LAVASS, are exponential equations $x = 2^y$. An LAVASS $V$ with two counters and $[\![L(V)]\!] = [\![x = 2^y]\!]$ is depicted in Figure 1. Control-states are depicted as circles and transitions as arrows between them. The vector before the colon is the alphabet symbol read. For instance, the transition from $q_0$ to $q_1$ reads the alphabet symbol $(1, 0) \in \{0, 1\}^2$. After a colon, we display the counter operations when reading the alphabet symbol, the operation on the first counter is displayed on the top and the operation on the second counter on the bottom. Here and subsequently, for presentational convenience, `id` is the identity function $x \mapsto x$, and `×2` and `×2+1` are the functions $x \mapsto 2x$ and $x \mapsto 2x + 1$, respectively. Thus, the transition from $q_0$ to $q_1$ applies the identity function on the first counter, and the function $x \mapsto 2x$ on the second counter.

The idea behind the gadget in Figure 1 is as follows. For an example, suppose that $y = 5$, then $x = 32$, and in binary the sequence of digits of $x$ and $y$ looks as follows:

$$\left[\begin{smallmatrix}x\\y\end{smallmatrix}\right] = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\cdots\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$$

Since $x = 2^y$, we have that $x \in 0^*10^*$, and the number of trailing zeros of $x$ is equal to the value of $y$. Thus, once a 1 in the binary representation of $x$ has been read, the first counter in the gadget of Figure 1 keeps incrementing and counts the number of trailing zeros of $x$. At the same time, the second counter in the gadget of Figure 1 keeps the value 0 until it reads the first 1 of the binary expansion of $y$, since $2 \cdot 0 = 0$. It then computes the value of $y$ in binary on the second counter by multiplying the value of the second counter by 2 when reading a zero, and multiplying by two and adding one when reading a one. The LAVASS in Figure 1 only accepts when the first and the second counter have the same value, i.e., when the number of trailing zeros of the binary expansion of $x$ equals the value of $y$, as required.

A closer look at the gadget constructed in Figure 1 reveals a number of important structural properties:

**(i)** all operations performed on the first counter are either the identity map `id` or increments `++`;
**(ii)** all operations performed on the second counter are affine updates `×2` and `×2+1`;
**(iii)** once the first counter gets incremented on a run, it gets incremented at every subsequent transition; and
**(iv)** only counter configurations in which the value of the first counter equals the value of the second counter are accepted.

Those properties are crucial to obtain decidability of (generalised) existential Semënov arithmetic.

▶ **Definition 5.** *An LAVASS is* restricted *if and only if it has an even number of $2d$ counters called $x_i, y_i$, $1 \le i \le d$, such that every counter pair $(x_i, y_i)$ adheres to the above Properties (i)–(iv), where $x_i$ is regarded as the first counter and $y_i$ as the second counter and the set of final counter values is defined by $\Phi \equiv \bigwedge_{1 \le i \le d} x_i = y_i$.*

For convenience, when referring to the counters in a pair, we subsequently refer to the first counter as its *$x$-counter* and to the second counter as its *$y$-counter*. We will usually write $m$ for the value of the $x$-counter and $n$ for the value of the $y$-counter. The following lemma is an immediate result of Figure 1 together with the previous discussion.

▶ **Lemma 6.** *There is a fixed restricted LAVASS $V$ of dimension two such that $L(V)$ is zero closed and $[\![L(V)]\!] = [\![x = 2^y]\!]$.*

Next, by following the constructions in Section 2.4 and noting that the counters of different LAVASS are simulated independently we obtain the following result.

▶ **Proposition 7.** *The languages of restricted LAVASS are closed under union, intersection, projection, and inverse projections.*

Thus, we are ready to present the central lemma of this section.

▶ **Lemma 8.** *Consider a positive conjunctive formula of Semënov arithmetic*

$$\Phi(\boldsymbol{x}) \equiv A \cdot \boldsymbol{x} = \boldsymbol{b} \wedge \bigwedge_{i \in I} x_i = 2^{y_i},$$

*where $A \in \mathbb{Z}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, $I$ is a finite index set, and $x_i$ and $y_i$ are variables from $\boldsymbol{x}$. There is a restricted LAVASS $V$ of dimension $2|I|$ and of size $(\|A\|_{1,\infty} + \|\boldsymbol{b}\|_\infty + 2)^{O(m+|I|)}$ such that $[\![L(V)]\!] = [\![\Phi]\!]$.*

This lemma follows from the combination of Lemma 4, Lemma 6, and Proposition 7.

Finally, for technical convenience, we assume that for a restricted LAVASS, we have $|Q| \geq 2$. This is with no loss of generality, since if $|Q| = 1$ then deciding non-emptiness is trivial (the restricted LAVASS has non-empty language if and only if the only control state is accepting).

The next sections will be devoted to the proof of the main result of this paper on restricted LAVASS.

▶ **Proposition 9.** *Language emptiness of a restricted LAVASS $V$ with $2d$ counters is decidable in* NSPACE$(|V| \cdot 2^{O(d)})$.

This proposition enables us to prove Theorem 1, by appealing to Lemma 8. Further details are relegated to Appendix A.2.

## 4 Certificates witnessing non-emptiness of restricted LAVASS

We now show that language emptiness for restricted LAVASS is decidable in exponential space. Clearly, this problem reduces to deciding whether a given restricted LAVASS has an accepting run, but witnessing runs may be of non-elementary length. To overcome this problem, we define an abstraction for configurations of restricted LAVASS. Abstract configurations store residue classes of counter values, as well as some further information that is required to witnesses the existence of concrete accepting runs. Before giving the formal definition, we provide some high level intuition that leads to our definition of abstract configurations. Next, we introduce reachability certificates, which are abstract runs with certain further properties. We argue that the existence of *witnessing certificates*, which are special kinds of reachability certificates witnessing that the language of an LAVASS is non-empty, are decidable in EXPSPACE. The last two sections then establish that witnessing certificates actually witness non-emptiness of restricted LAVASS.

## 4.1 Key observations

Given a *restricted* LAVASS $V$ in dimension $d$, assuming that $L(V) \neq \emptyset$, there is a run $\pi$ from an initial configuration $c$ to a final configuration $c'$. With no loss of generality, throughout this section, we assume that $\mathit{val}(c', x_i) \geq \mathit{val}(c', x_{i+1}) > 0$ for all $1 \leq i < d$. In particular, this implies that every counter gets incremented at least once along a path witnessing non-emptiness.

Our first observation is that if, along $\pi$, a counter $y_i$ achieves the first time a non-zero value by taking a `×2+1` labeled transition, then the length of the remaining segment of $\pi$ is bounded by $O(\log(m_i + 1))$, where $m_i$ is the value of counter $x_i$ before the transition is taken. The reason is that, once $y_i$ has non-zero value, its value is at least doubling for every following transition taken. Hence if $\pi$ is "long" then along $\pi$ there will be loops incrementing a counter $x_i$ before the corresponding $y_i$ achieves non-zero value.

In the latter scenario, we may actually, subject to some bookkeeping, discard concrete values of $x_i$ and $y_i$ and only store their residue classes modulo $\ell_i$, where $\ell_i$ is the length of the first loop incrementing $x_i$ along $\pi$. In particular, if we are given a non-accepting run $\pi'$ such that $\mathit{val}(\pi', x_i) \equiv \mathit{val}(\pi', y_i) \bmod \ell_i$ and $\mathit{val}(\pi', x_i) < \mathit{val}(\pi', y_i)$ then $\pi'$ can be turned into a run $\pi''$ where $\mathit{val}(\pi'', x_i) = \mathit{val}(\pi'', y_i)$ by iterating the loop of length $\ell_i$.

There are, however, some further subtleties that need to be taken care of. Consider the segment $\pi'$ of $\pi$ between the first transition labeled by `++` on $x_i$ and the first transition labeled by `++` on $x_{i+1}$. If $\pi'$ contains no loop then we are in a situation where the first loop incrementing $x_i$ is also the first loop incrementing $x_{i+1}$. This means that the values of $x_i$ and $x_{i+1}$ get paired together, and hence, for an accepting run, also the values of $y_i$ and $y_{i+1}$ are paired together. In our approach, we deal with such circumstances by introducing so-called *$y$-constraints*. A $y$-constraint of the form $y_i - y_{i+1} = \delta_i$ for some constant $\delta_i \in \mathbb{N}$ asserts that the counters $y_{i+1}$ and $y_i$ must eventually have constant difference $\delta_i$ along a run.

Otherwise, if $\pi'$ above contains a loop, the difference between the values of $x_i$ and $x_{i+1}$ is not necessarily constant, but lower-bounded by the length $\delta_i$ of the loop-free sub path of $\pi'$. Thus, in an accepting run, the difference between $y_i$ and $y_{i+1}$ must also be at least $\delta_i$, which is asserted by a $y$-constraint of the form $y_i - y_{i+1} \geq \delta_i$.

## 4.2 Abstract configurations for restricted LAVASS

Our decision procedure for non-emptiness of restricted LAVASS is based on reducing this problem to a reachability problem in a carefully designed finite-state abstraction of the state-space of LAVASS. Throughout this section, let $V = \langle Q, 2d, \Sigma, \Delta, \lambda, q_0, F, \Phi\rangle$ be a restricted LAVASS. We first define the state space of the abstracted LAVASS.

▶ **Definition 10.** *An abstract configuration is a tuple*

$$\alpha = (q, m_1, n_1, \ldots, m_d, n_d, u_1, u_2, \ldots, u_{d-1}, \ell_1, \ldots, \ell_d) \\ \in Q \times (\mathbb{N} \cup \{\bot\})^{2d} \times \mathbb{N}^{d-1} \times (\mathbb{N} \cup \{\top\})^d,$$

*such that* $m_i, n_i \in [0, 2dM_i] \cup \{\bot\}$ *and* $u_i \in [0, U_i]$ *and* $\ell_i \in [0, M_i - 1] \cup \{\top\}$*, where*
- $M_i := \lfloor |Q|^{((1/8)\cdot 32^{i-1}+1)} \rfloor$*; and*
- $U_i := |Q|^{(32^{i-1}+4)}$.

The idea is that $m_i, n_i$ store the residue classes modulo $\ell_i$ of the counter pair $x_i, y_i$ respectively, where the value $\top$ for $\ell_i$ acts as an indicator that we are storing actual values and not residue classes. The value $\bot$ for some $x_i$ or $y_i$ indicates that the counter has not yet been initialised. If for an update function $f$, $f =$ `++` or $f =$ `×2+1` then $f(\bot) := 1$; otherwise $f(\bot) := \bot$, and we stipulate that $\bot \bmod n = \bot$. The value of $u_i$ in an abstract configuration carries the current difference between the value of the counters $y_i$ and $y_{i+1}$. This difference is potentially unbounded, however for our purposes it suffices to only store its value if it is less than $U_i$, and to indicate the fact that it is at least $U_i$ by the value $u_i = U_i$.

We denote the (finite) set of all abstract configurations of $V$ by $A(V)$. Let us now define a transition relation $\dot{\rightarrow} \subseteq A(V) \times \Delta \times A(V)$ such that $\alpha \xrightarrow{t} \alpha'$, $t = (q, a, q') \in \Delta$ if and only if:

- $\alpha = (q, m_1, n_1, \ldots, u_{d-1}, \ell_1, \ldots, \ell_d)$ and $\alpha' = (q', m'_1, n'_1, \ldots, u'_{d-1}, \ell_1, \ldots, \ell_d)$;
- $\lambda(t) = (f_{x_1}, f_{y_1}, \ldots, f_{x_d}, f_{y_d})$;
- $f_{x_i} = \texttt{++}$ for all $i$ such that $m_i \neq \bot$;
- if $\ell_i \neq \top$, $m'_i = f_{x_i}(m_i) \bmod \ell_i$ and $n'_i = f_{y_i}(n_i) \bmod \ell_i$;
- if $\ell_i = \top$, $m'_i = f_{x_i}(m_i)$ and $n'_i = f_{y_i}(n_i)$; and
- for all $i \in \{1, \ldots, d-1\}$,

$$u'_i = \begin{cases} min(2u_i + 1, U_i) & \text{if } f_{y_i} = \texttt{×2+1}, f_{y_{i+1}} = \texttt{×2} \\ min(2u_i, U_i) & \text{if } f_{y_i} = f_{y_{i+1}} \\ min(2u_i - 1, U_i) & \text{if } f_{y_i} = \texttt{×2}, f_{y_{i+1}} = \texttt{×2+1}. \end{cases}$$

Assuming that the value of $y_i$ is at least the value of $y_{i+1}$, which we will always ensure, the definition of how to update $u_i$ ensures that it exactly stores the difference $y_i - y_{i+1}$ unless the difference becomes too large, in which case it is levelled off at $U_i$.

An abstract configuration path is a sequence of abstract configurations and transitions of the form $R = \alpha_1 \xrightarrow{t_1} \alpha_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-1}} \alpha_n$.

Given two consecutive $y$-counters $y_i, y_{i+1}$ and $\delta_i \in \mathbb{N}$, we say that $y_i - y_{i+1} = \delta_i$ and $y_i - y_{i+1} \geq \delta_i$ are *$y$-constraints*. Let $Y$ be a set of $y$-constraints, an abstract configuration $\alpha = (q, m_1, n_1, \ldots, u_1, \ldots, u_{d-1}, \ell_1, \ldots, \ell_d)$ *respects* $Y$ whenever

- $u_i \geq \delta_i$ for all constraints of type $y_i - y_{i+1} \geq \delta_i$ in $Y$,
- and $u_i < U_i$ and $u_i = \delta_i$ for all constraints $y_i - y_{i+1} = \delta_i$ in $Y$.

We say that $\alpha_f$ is a *final abstract configuration respecting* $Y$ whenever $q \in F$, $m_i = n_i$ for all $1 \leq i \leq d$, and $\alpha_f$ respects $Y$.

## 4.3 Witnessing certificates

While any concrete accepting run of an LAVASS gives rise to an abstract configuration path ending in an accepting abstract configuration, the converse does not hold. This motivates the introduction of reachability and witnessing certificates, which are special abstract configuration paths that carry further information that eventually enables us to derive from a witnessing certificate a concrete accepting run of an LAVASS.

A *reachability certificate* is a tuple $(R, X, Y, L)$ such that $R = \alpha_1 \xrightarrow{t_1} \alpha_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-1}} \alpha_n$ is an abstract configuration path, and $X, Y, L \colon \{1, \ldots, d\} \to \{1, \ldots, n\}$. Here, $X(i)$ and $Y(i)$ indicate the position where the $x_i$-counter and $y_i$-counter obtain a value different from $\bot$ for the first time. Moreover, $L(i)$ is the position where a loop of length $\ell_i$ can be found. Formally, $(R, X, Y, L)$ is required to have the following properties:

**(a)** $\alpha_1 = (q_0, \bot, \ldots, \bot, 0, \ldots, 0, \ell_1, \ldots, \ell_d)$ and if $\ell_i = \top$ then $\ell_j = \top$ for all $i < j \leq d$;
**(b)** $\lambda(x_i, t_{X(i)-1}) = \texttt{++}$ and $\lambda(y_i, t_{Y(i)-1}) = \texttt{×2+1}$ for all $1 \leq i \leq d$;
**(c)** $\lambda(x_i, t_j) = \texttt{id}$ for all $1 \leq j < X(i) - 1$;
**(d)** $\lambda(y_i, t_j) = \texttt{×2}$ for all $1 \leq j < Y(i) - 1$;
**(e)** $X, Y, L$ are monotonic; and
**(f)** for all $1 \leq i \leq d$, if $\ell_i \neq \top$ then
  - $X(i) \leq L(i) < Y(i)$, and
  - there is a simple $\alpha_{L(i)}$-loop $\alpha_{L(i)} \xrightarrow{t'_1} \alpha'_2 \xrightarrow{t'_2} \cdots \xrightarrow{t_{\ell_i - 1}} \alpha'_{\ell_i - 1} \xrightarrow{t'_{\ell_i}} \alpha_{L(i)}$ of length $\ell_i$.

Those conditions can be interpreted as follows. Condition (a) asserts that the certificate starts in an initial abstract configuration. We require that $\top$ monotonically propagates since the absence of a loop for counter $x_i$ implies that the remainder of a path is short, hence we can afford to subsequently store actual counter values and not residue classes. Conditions (b), (c) and (d) assert that $X(i)$ and $Y(i)$ are the first position where the counters $x_i, y_i$ hold a value different from $\bot$. Condition (e) states that the counters $x_{i+1}$, $y_{i+1}$ do not carry a value different from $\bot$ before the counters $x_i$ and $y_i$, respectively. Condition (f) implies that, if $\ell_i \neq \top$ then between the first update for counter $x_i$ and the first update for counter $y_i$ there is a position $L(i)$ where we can find a loop in the abstract configurations of length $\ell_i$. Notice that if $x_j = \bot$ or $y_j = \bot$ in $\alpha_{L(i)}$ then $x_j$ and $y_j$ remain to hold $\bot$ along this loop, i.e., this loop does not update counters that have not been initialised already.

Given $R$, the set of $y$-constraints induced by $R$ is the smallest set containing

- $y_i - y_{i+1} \geq \delta_i$, where $\delta_i := X(i+1) - X(i)$ if there is a $j$ such that $X(i) \leq L(j) < X(i+1)$; and
- otherwise $y_i - y_{i+1} = \delta_i$, where $\delta_i := X(i+1) - X(i)$,

for all $1 \leq i < d$ such that $\ell_i \neq \top$.

We introduce some further notation. Given a reachability certificate $(R, X, Y, L)$, we denote by $\pi(R)$ the run corresponding to $R$ in the configuration graph of $V$, with the initial configuration $(q_0, 0, 0, \ldots, 0, 0)$. Given indices $1 \leq i \leq j \leq n$, we denote by $R[i, j]$ the segment $\alpha_i \xrightarrow{t_i} \alpha_{i+1} \cdots \xrightarrow{t_{j-1}} \alpha_j$ of $R$, and by $R[i] := \alpha_i$. We say that $R$ is a *witnessing certificate* if, for $a \leq d$ being the largest index such that $\ell_a \neq \top$:

- $R[1, Y(a)]$ is a simple path and $n - Y(a) \leq 2dM_{d+1}$;
- $\alpha_n$ is a final abstract configuration respecting the set of induced $y$-constraints; and
- $\mathit{val}(\pi(R), x_a) \leq \mathit{val}(\pi(R), y_a)$.

Sometimes we will speak of witnessing certificates *restricted* to a set of counters. By that we mean a witnessing certificates where the relevant Conditions (a)–(f) are only required for that set of counters.

In the next section, we prove the following theorem that will enable us to give a proof for Proposition 9.

▶ **Theorem 11.** *The language of a restricted LAVASS $V$ is non-empty if and only if there exists a witnessing certificate for $V$.*

## 5 Witnessing certificates witness non-emptiness of restricted LAVASS

In this section we prove Theorem 11. The proof is split into the two directions. First, we argue that the existence of a witnessing certificate for an LAVASS implies that the language of the LAVASS is non-empty. Subsequently, we show the converse direction.

▶ **Proposition 12.** *If there exists a witnessing certificate for a restricted LAVASS $V$ then $L(V) \neq \emptyset$.*

The idea behind the proof of Proposition 12 is that, from a witnessing certificate $(R, X, Y, L)$ of an LAVASS $V$, we obtain a sequence of runs of $V$ such that the final run in that sequence is an accepting run of $V$. Initially, we obtain a run that ends in a configuration where the counters are in a congruence relation. We then carefully pump the simple loops pointed to by $L$, beginning from the last counter and working towards the first. The formal proof is presented in Appendix A.3.

We now turn towards the converse direction and show that we can obtain a witnessing certificate from a run witnessing non-emptiness.

▶ **Proposition 13.** *If $L(V) \neq \emptyset$ for a restricted LAVASS $V$ then there exists a witnessing certificate for $V$.*

We begin by defining a function that turns a configuration from $C(V)$ into an abstract configuration. This function is parameterised by $\ell_1, \ldots, \ell_d \in \mathbb{N}_+ \cup \{\top\}$:

$$f_V((q, m_1, n_1, \ldots, m_d, n_d), \ell_1, \ldots, \ell_d) := (q, m_1 \circ \ell_1, n_1 \circ \ell_1, \ldots, m_d \circ \ell_d, n_d \circ \ell_d, \\ min(n_1 - n_2, U_1), \ldots, min(n_{d-1} - n_d, U_{d-1}), \ell_1, \ldots, \ell_d) .$$

Here, $m \circ \ell := \bot$ if $m = 0$; $m \circ \ell := m \bmod \ell$ if $\ell \in \mathbb{N}_+$; and $m \circ \ell := m$ if $\ell = \top$. We lift the definition of $f_V$ to concrete runs $\pi$ in the natural way, and write $f_V(\pi, \ell_1, \ldots, \ell_d)$ for the resulting sequence of abstract configurations. Let $\pi = c_1 \xrightarrow{t_1} c_2 \cdots \xrightarrow{t_{n-1}} c_n$ be a run witnessing $L(V) \neq \emptyset$. We show how to obtain a witnessing certificate $R$ from $\pi$. Without loss of generality, in $c_n$ we have $m_1 \geq m_2 \geq \ldots m_d > 0$.

To this end, we show how from the accepting run $\pi$ we can iteratively define a sequence $R_0, R_1, R_2 \ldots, R_d$ of abstract runs and identify the required $\ell_1, \ldots, \ell_d \in \mathbb{N}_+ \cup \{\top\}$ and $X, Y, L$ such that $(R_d, X, Y, L)$ is a reachability certificate. Let $X(i) := j$ such that $j$ is the first position in $\pi$ where the value of counter $x_i$ is non-zero; analogously define $Y(i)$ to be the first position where the value of $y_i$ is non-zero. Clearly, $X, Y$ are monotonic and $X(i) \leq Y(i)$, for all $1 \leq i \leq d$. Otherwise, if a counter $y_i$ gets initialised before the counter $x_i$ in $\pi$, it must be the case that $n_i > m_i$ in $c_n$ and therefore $c_n$ cannot be an accepting configuration.

Recall that $\pi$ has length $n$. In our proof, the subsequent technical lemma will allow us to conclude that, if for a counter pair $x_i, y_i$ the $y_i$ counter gets updated shortly after the $x_i$ counter then the run will end shortly after and counter pairs $x_j, y_j$ for $j \geq i$ will consequently have small values.

▶ **Lemma 14.** *If $Y(i) - X(i) \leq dM_i$ for some $1 \leq i \leq d$ then $n - Y(i) < dM_i$, so $m_j, n_j \leq 2dM_i$ in $c_n$ for all $i \leq j \leq d$.*

**Proof.** We have that $Y(i) - X(i) \leq dM_i$ implies that $val(\pi[1, Y(i)], x_i) \leq dM_i + 1$, and since $val(\pi[1, Y(i) + k], y_i) \geq 2^k$ we get that:

- $val(\pi, y_i) \geq 2^{n-Y(i)}$; and
- $val(\pi, x_i) \leq dM_i + n - Y(i) + 1$.

Assume that $n - Y(i) \geq dM_i$. Then, $2^{n-Y(i)} - (dM_i + n - Y(i) + 1) > 2^{n-Y(i)} - (2n - 2Y(i) + 1) > 0$, if $n - Y(i) \geq 3$. However, $\pi$ is an accepting path, so $val(\pi, x_i) = val(\pi, y_i)$, and we get a contradiction. Thus, we must have that $n - Y(i) < dM_i$ which implies that $val(\pi, x_i) \leq 2dM_i$, so $m_i = n_i \leq 2dM_i$ and for any $j$, $i < j \leq d$, $m_j \leq m_i$ and $n_j \leq n_i$, so $m_j, n_j < 2M_i$ in $c_n$, for all $i \leq j \leq d$ since $\pi$ is an accepting path. ◀

Let $R_0 := f_V(\pi, 1, 1, \ldots, 1)$. Note that $R_0$ together with $X$ and $Y$ as defined above adhere to Conditions (a)–(e) of reachability certificates.

Suppose $R_{i-1}$ and $\ell_1, \ldots, \ell_{i-1}$ have been constructed. If $i > 1$, and $L(i-1) \geq X(i)$ or $\ell_{i-1} = \top$ then we choose $\ell_i := \ell_{i-1}$, $L(i) = L(i-1)$ and $R_i := f_V(\pi, \ell_1, \ldots, \ell_i, 1, \ldots, 1)$. Otherwise, we distinguish two cases.

- $Y(i) - X(i) < dM_i$: we choose $\ell_i := \top$ and $L(i) := X(i)$.
- $Y(i) - X(i) \geq dM_i$: then there is a segment in $R_{i-1}[X(i), Y(i)]$ of length greater than $M_i$ on which no $x$-counter has its first ++ transition. Let $N_i$ be the number of different abstract configurations on this segment. Since $\ell_{i-1} \neq \top$ we know that $m_j, n_j$ can take at most $M_j$ different values for all $1 \leq j < i$, as they can either be $\bot$ or a residue class

modulo $M_j$. Also, for all $i \le j \le d$ the values of $m_j, n_j$ have a constant value, either 0 or $\bot$, on this segment, and $u_i = \cdots = u_d = 0$ in all abstract configurations of this segment. So

$$\begin{aligned}
N_i &\le |Q| \prod_{1 \le j < i} M_j^2 \cdot U_j \\
&\le |Q| \prod_{1 \le j < i} |Q|^{(1/4)\cdot 32^{j-1}+2+32^{j-1}+4} \\
&\le |Q|^{(1/3968)(5\cdot 32^i - 23968)+6i+1} \\
&< |Q|^{(1/8)\cdot 32^{i-1}+1} \\
&= M_i
\end{aligned}$$

By the pigeonhole principle, there is a smallest $k$, $X(i) \le k < Y(i)$, $\ell < M_i$, and a simple loop $\alpha_k \xrightarrow{t_k} \cdots \xrightarrow{t_{k+\ell}} \alpha_{k+\ell+1} = \alpha_k$ in $R_{i-1}$. We choose $L(i) := k$, $\ell_i := \ell$ and let $R_i := f_V(\pi, \ell_1, \dots, \ell_i, 1, \dots, 1)$.

By construction, $(R_d, X, Y, L)$ is a reachability certificate. It remains to turn it into a witnessing certificate. In particular, this requires to remove loops from $R_d$, to ensure that the final segment of $R_d$ is short, and to establish that $R_d$ is consistent with the implied $y$-constraints.

Let $R := R_d = f_V(\pi, \ell_1, \dots, \ell_d)$ and $a$ be the largest index such that $\ell_a \ne \top$. In order to make $R$ loop-free, we iterate the following process:

- Identify the first simple loop $\alpha_k \xrightarrow{t_k} \cdots \xrightarrow{t_{k+\ell}} \alpha_{k+\ell+1}$ in $R[1, Y(a)]$ and replace it by $\alpha_k$; observe that for $I := \{k+1, \dots, k+\ell\}$, we have $I \cap \{X(i), Y(i), L(i) : 1 \le i \le d\} = \emptyset$ since $\alpha_{X(i)-1} \xrightarrow{t_{X(i)-1}} \alpha_{X(i)}$ occurring in $R_d$ means that $x_i$ has value $\bot$ in $\alpha_{X(i)-1}$ and a value different from $\bot$ in $\alpha_{X(i)}$, and thus $\alpha_{X(i)}$ cannot be part of a loop; the same argument applies to any $Y(i)$. Finally, since $L(i)$ was chosen as the index of the first configuration of the first cycle appearing after $X(i)$, we have $L(i) \notin I$ for all $1 \le i \le d$ as well.
- Update $X, Y, L$ such that for all $i$ such that $X(i) > k$, $X(i) := X(i) - \ell$, and analogously $Y(i) := Y(i) - \ell$ and $L(i) := L(i) - \ell$ for the respective $i$.

This process guarantees that $R[1, Y(a)]$ is loop-free. It is easy to verify that $(R, X, Y, L)$ obtained in this way is a reachability certificate and that the last abstract configuration of $R$ is accepting.

We now show that the $y$-constraints induced by $R$ are valid in the final configuration of $R$. To this end, we first show that for all $1 \le i \le d$ such that $\ell_i \ne \top$, $X(i+1) - X(i) < U_i$. Consider the simple path $\alpha_{X(i)} \xrightarrow{t_{X(i)}} \alpha_{X(i)+1} \xrightarrow{t_{X(i)+1}} \cdots \xrightarrow{t_{X(i+1)-1}} \alpha_{X(i+1)}$. If $Y(i) \ge X(i+1)$ then clearly $X(i+1) - X(i) \le N_i < U_i$, where $N_i$ is defined as above. Otherwise, there is a $k \in \mathbb{N}$ such that the path decomposes as

$$\alpha_{X(i)} \xrightarrow{t_{X(i)}} \cdots \alpha_{Y(i)} \xrightarrow{t_{Y(i)}} \cdots \alpha_{Y(i)+k} \xrightarrow{t_{Y(i)+k}} \cdots \xrightarrow{t_{X(i+1)-1}} \alpha_{X(i+1)}$$

and

- $u_i = 0$ in all abstract states $\alpha_j$ with $X(i) \le j \le Y(i)$;
- $u_i = U_i$ in all abstract states $\alpha_j$ with $Y(i) + k \le j \le X(i+1)$; and
- $k \le \log U_i$.

Thus, the maximum length of $R[X(i), X(i+1)]$ is bounded by:

$$\begin{aligned}
&N_i \cdot M_i + \log U_i + N_i \cdot 2M_i \\
&\leq 2 \cdot M_i^3 + \log U_i \\
&\leq |Q|^{(3/8)\cdot 32^{i-1}+4} + |Q|^{5(i-1)+1} + 4|Q| \\
&< |Q|^{32^{i-1}+4} \\
&= U_i
\end{aligned}$$

We can now show that $R$ respects the induced $y$-constraints. Fix some $1 \leq i \leq d$ such that $\ell_i \neq \top$. We distinguish two cases.

- There is no $1 \leq j \leq a$ such that $X(i) \leq L(j) \leq X(i+1)$. Thus, we know that $y_i - y_{i+1} = \delta_i$ is in the set of induced $y$-constraints. Also, $val(\pi, y_i) - val(\pi, y_{i+1}) = val(\pi, x_i) - val(\pi, x_{i+1}) = X(i+1) - X(i) = \delta_i$ since we did not remove any abstract loops on the segment of $R_d$ between the first **++** update for $x_i$ and the first **++** update for $x_{i+1}$. Finally, since $\delta_i < U_i$ by the above argument, we conclude that $u_i = \delta_i$ in the last abstract configuration $R[n]$ of $R$.
- Otherwise, $X(i) \leq L(i) \leq Y(i)$, so $y_i - y_{i+1} \geq \delta_i$ is in the set of induced $y$-constraints. However, $val(\pi, y_i) - val(\pi, y_{i+1}) = val(\pi, x_i) - val(\pi, x_{i+1}) \geq X(i+1) - X(i) = \delta_i$ and again because $\delta_i < U_i$ we can conclude that $u_i \geq \delta_i$ in $R[n]$.

This establishes that the $y$-constraints are satisfied. Let $n$ be the index of the last abstract configuration of $R$. For the final step, we now argue that $val(\pi(R), x_a) \leq val(\pi(R), y_a)$ and $n - Y(a) \leq 2dM_{d+1}$. We make a case distinction:

- $a = d$: Note that $val(\pi, x_d) = val(\pi, y_d)$. Since we only remove loops from $R_d[1, Y(d)]$, we have that $val(\pi(R), x_d) \leq val(\pi(R), y_d)$. If $n - Y(d) \leq 2dM_{d+1}$ we are done with $(R, X, Y, L)$ as a witnessing certificate. Otherwise, assume $n - Y(d) > 2dM_{d+1}$. This implies that the path $R[Y(d), n]$ must contain at least one simple loop. Consider iterating the following process:
  - remove the first simple loop from $R[Y(d), n]$ and update $n := n - \ell$, where $\ell$ is the length of the simple loop that was removed; and
  - stop if $n - Y(d) \leq 2dM_{d+1}$.

  We argue that, $n - Y(d) \geq M_d^2$. Let $R'$ and $n'$ be the previous values of $R, n$ before the last iteration. It must be that $n' > 2dM_{d+1}$ and since the length of any simple loop of $R'[Y(d)+1, n']$ is bounded by $M_{d+1}$, we get that $n - Y(a) \geq M_{d+1} \geq M_d^2$. Note that $Y(d) - X(d) \leq M_d \cdot |Q| \cdot \prod_{1 \leq j < d} M_j^2 \cdot U_j \leq M_d^2$, so $val(\pi(R[1, Y(d)]), x_d) \leq M_d^2$. It must be then the case that $val(\pi(R), x_d) \leq val(\pi(R), y_d)$.
- $a < d$: we know $n - Y(a) \leq 2dM_{d+1}$ by Lemma 14. Moreover, we must have that $val(\pi(R), x_a) \leq val(\pi(R), y_a)$ since $val(\pi, x_a) = val(\pi, y_a)$ and we do not remove loops after the counter $y_a$ is incremented.

We are now ready to prove Proposition 9, i.e., show that language emptiness for restricted LAVASS can be decided in NSPACE$(|V| \cdot 2^{O(d)})$.

**Proof of Proposition 9.** Clearly, an abstract configuration can be stored in space $|V| \cdot 2^{O(d)}$. An NEXPSPACE algorithm can hence non-deterministically choose an initial configuration and non-deterministically verify that it leads to a final abstract configuration along a path that is a witnessing certificate. To this end, the algorithm computes the set of induced $y$-constraints on-the-fly while guessing the reachability certificate, and verifies them in the last configuration. Note that the $y$-constraints can be stored in space $|V| \cdot 2^{O(d)}$. Finally, the requirement $\mathit{val}(\pi(R), x_a) \leq \mathit{val}(\pi(R), y_a)$ can also be verified in exponential space since we require that $R[1, Y(a)]$ is a simple path and $n - Y(a) \leq 2M_{d+1}$. ◀

## 6 A decidable class of string constraints

In this section, we show that a certain fragment of string constraints, whose decidability status has been left open in the literature, can be reduced in logarithmic space to generalised Semënov arithmetic, and is hence decidable in EXPSPACE. This demonstrates an important application of our results on generalised Semënov arithmetic, with deep connections to solving string constraints in practice, which has been one of the motivations for our work.

Let $\Sigma = \{0, 1\}$. The *theory of enriched string constraints* $T_{\text{lnc}}$ is the first-order theory of the two-sorted structure

$$\langle \Sigma^*, \mathbb{N}; \{w\}_{w \in \Sigma^*}, \cdot, \mathit{len}, \mathit{sn}, \{R_i\}_{i \in \mathbb{N}}, 0, 1, + \rangle,$$

where

- the binary function $\cdot$ over $\Sigma^*$ is the string concatenation operator,
- the unary function $\mathit{len} \colon \Sigma^* \to \mathbb{N}$ returns on input $w$ the length $|w|$ of $w$,
- the unary function $\mathit{sn} \colon \Sigma^* \to \mathbb{N}$ on input $u$ returns $[\![u]\!]$, and
- $R_0, R_1, \ldots \subseteq \Sigma^*$ is an enumeration of all regular languages.

The remaining predicates, constant and function symbols are defined in their standard semantics.

The above theory was introduced in [4], where an SMT solver addressing some fragments of this theory was defined, implemented, and compared to other state of the art solvers which can handle such string constraints. Extending [4], [3] presents in more details the motivation behind considering this theory and its fragments. More precisely, the authors of [3] analysed an extensive collection of standard real-world benchmarks of string constraints and extracted the functions and predicates occurring in them. The works [3, 4] focused on benchmarks that do not contain word equations, and the result of the aforementioned benchmark-analysis produced exactly the four functions and predicates mentioned above: *len*, *sn*, regular language membership, and concatenation of strings.

Complementing the practical results of [4], [3] showed a series of theoretical results regarding fragments of $T_{\text{lnc}}$. In particular, the existential theory of $T_{\text{lnc}}$ is shown to be undecidable. Moreover, [3] leaves as an open problem the question whether the existential theories of $T_{\text{REln}}$ and $T_{\text{REnc}}$, which drop the concatenation operator and length function, respectively, are decidable. From these two, the existential theories of $T_{\text{REln}}$ seems particularly interesting, as all instances from the benchmarks considered in the analysis [3] can be easily translated into a formula from this particular fragment of $T_{\text{REln}}$. Indeed, by the results reported in Table 1.b from [3], no instance contains both concatenation of strings and the *sn* function; moreover, the concatenation of strings, which appears only in formulas involving regular membership predicates and, in some cases, length function, can be easily removed in all cases by a folklore technique called automata splitting (see, e.g., [1]). Therefore, showing that the existential fragment of $T_{\text{REln}}$ is decidable actually shows that one can decide the satisfiability of all the instances from the standard benchmarks analysed in [3].

In this paper, we solve this open problem. By a reduction to generalised Semënov arithmetic, we can settle the decidability status of $T_{\text{REln}}$:

▶ **Theorem 15.** *The existential fragment of $T_{\text{REln}}$ is decidable in EXPSPACE.*

The idea underlying our proof is that we map a string $s$ to the number $[\![1s]\!]$. Note that we cannot directly treat strings in $\Sigma^*$ as natural numbers due to the possibility of leading zeros. This encoding enables us to treat strings as numbers and to implement the functions *sn* and *len* in generalised Semënov arithmetic as $sn(s,x) \iff \exists y. 2^y \le s \land s < 2^{y+1} \land x = s - 2^y$ and $len(s,x) \iff 2^x \le s \land s < 2^{x+1}$ respectively. A full proof is presented in Appendix A.4.

## 7 Conclusion

The main result of this article has been to show that the existential theory of generalised Semënov arithmetic is decidable in EXPSPACE. On a technical level, this result was obtained by showing that a restricted class of labelled affine VASS has an EXPSPACE-decidable language emptiness problem. The structural restrictions imposed on those restricted LAVASS are rather strong, though necessary to obtain a decidable class of LAVASS.

As an application of this main result, we showed that a highly relevant class of string constraints with length constraints is also decidable in EXPSPACE; the decidability of this class was the main problem left open in [3]. Furthermore, our decision procedure has a better complexity than the one reported in [15] for a fragment of generalised Semënov arithmetic.

An interesting aspect of our approach is that it establishes automaticity of the existential fragment of a logical theory that is different from traditional notions of automaticity, which are based on finite-state automata or tree automata over finite or infinite words and trees [5, 10], respectively. It would be interesting to better understand whether there are natural logical theories whose (existential) fragments are, say, Petri-net or visibly-pushdown automatic.

We have ignored algorithmic lower bounds throughout this article, but it would, of course, be interesting to see whether the upper bounds of the decision problems we considered in this article are tight. It is clear that generalised Semënov arithmetic is PSPACE-hard since it can readily express the DFA intersection non-emptiness problem, but this still leaves a considerable gap with respect to the EXPSPACE upper bound we established. In particular, the recent results of [2] showing an NEXP upper bound for the existential fragment of Semënov arithmetic suggest that, if an EXPSPACE lower bound for existential generalised Semënov arithmetic is possible, it will require the use of regular predicates.

## References

1 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Yu-Fang Chen, Lukás Holík, Ahmed Rezine, Philipp Rümmer, and Jari Stenman. Norn: An SMT solver for string constraints. In *Proc. Computer Aided Verification, CAV*, volume 9206 of *Lect. Notes Comp. Sci.*, pages 462–469, 2015. doi:10.1007/978-3-319-21690-4_29.

2 Michael Benedikt, Dmitry Chistikov, and Alessio Mansutti. The Complexity of Presburger Arithmetic with Power or Powers. In *International Colloquium on Automata, Languages, and Programming, ICALP*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 112:1–112:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.112.

3 Murphy Berzish, Joel D. Day, Vijay Ganesh, Mitja Kulczynski, Florin Manea, Federico Mora, and Dirk Nowotka. Towards more efficient methods for solving regular-expression heavy string constraints. *Theor. Comput. Sci.*, 943:50–72, 2023. doi:10.1016/j.tcs.2022.12.009.

4 Murphy Berzish, Mitja Kulczynski, Federico Mora, Florin Manea, Joel D. Day, Dirk Nowotka, and Vijay Ganesh. An SMT solver for regular expressions and linear arithmetic over string length. In *Computer Aided Verification, CAV*, volume 12760 of *Lect. Notes Comp. Sci.*, pages 289–312, 2021. doi:10.1007/978-3-030-81688-9_14.

5 Achim Blumensath and Erich Grädel. Automatic structures. In *Logic in Computer Science, LICS*, pages 51–62. IEEE Computer Society, 2000. doi:10.1109/LICS.2000.855755.

6 Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and $p$-recognizable sets of integers. *Bull. Belg. Math. Soc. Simon Stevin*, 1(2):191–238, 1994. doi:10.36045/bbms/1103408547.

7 Gregory Cherlin and Françoise Point. On extensions of Presburger arithmetic. In *Proc. 4th Easter Model Theory conference, Gross Köris*, pages 17–34, 1986.

8 Alain Finkel, Stefan Göller, and Christoph Haase. Reachability in register machines with polynomial updates. In *Proc. Mathematical Foundations of Computer Science, MFCS*, volume 8087 of *Lect. Notes Comp. Sci.*, pages 409–420. Springer, 2013. doi:10.1007/978-3-642-40313-2_37.

9 Florent Guépin, Christoph Haase, and James Worrell. On the existential theories of Büchi arithmetic and linear $p$-adic fields. In *Proc. Symposium on Logic in Computer Science, LICS*, pages 1–10, 2019. doi:10.1109/LICS.2019.8785681.

10 Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *Logical and Computational Complexity, LCC*, volume 960 of *Lect. Notes Comp. Sci.*, pages 367–392. Springer, 1995. doi:10.1007/3-540-60178-3_93.

11 Françoise Point. On the expansion of $(\mathbb{N}, +, 2^x)$ of Presburger arithmetic. Unpublished manuscript, 2010. URL: https://u.osu.edu/friedman.8/files/2014/01/OAppB072710-1mnlwyn.pdf.

12 Julien Reichert. *Reachability games with counters: decidability and algorithms.* PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2015.

13 A. L. Semënov. On certain extensions of the arithmetic of addition of natural numbers. *Mathematics of the USSR-Izvestiya*, 15(2):401, 1980. doi:10.1070/IM1980v015n02ABEH001252.

14 Pierre Wolper and Bernard Boigelot. On the construction of automata from linear arithmetic constraints. In *Proc. Tools and Algorithms for the Construction and Analysis of Systems, TACAS*, pages 1–19, 2000. doi:10.1007/3-540-46419-0_1.

15 Hao Wu, Yu-Fang Chen, Zhilin Wu, Bican Xia, and Naijun Zhan. A decision procedure for string constraints with string/integer conversion and flat regular constraints. *Acta Informatica*, October 2023. doi:10.1007/s00236-023-00446-4.

## A Appendix

### A.1 Closure properties of LAVASS languages

Let $V_i = \langle Q_i, d_i, \Sigma, \Delta_i, \lambda_i, q_0^{(i)}, F_i, \phi_i \rangle$, $i \in \{1, 2\}$, be two LAVASS.

▶ **Proposition 16.** *The languages of LAVASS are closed under union and intersection.*

Closure under union is trivial since we allow for non-determinism. To show closure under intersection, we define the LAVASS $V := (Q', d_1 + d_2, \Sigma, \Delta', \lambda', q_0', F', \phi')$ such that

- $Q' := Q_1 \times Q_2$,
- $((q_1, q_2), a, (r_1, r_2)) \in \Delta'$ if and only if $(q_1, a, r_1) \in \Delta_1$ and $(q_2, a, r_2) \in \Delta_2$,
- $\lambda'((q_1, q_2), a, (r_1, r_2)) := (\lambda_1(q_1, a, r_1), \lambda_2(q_2, a, r_2))$,
- $q_0'' := (q_0^{(1)}, q_0^{(2)})$,
- $F' := F_1 \times F_2$, and
- $\phi$ is the conjunction of $\phi_1$ and $\phi_2$, with counters renamed accordingly.

▶ **Lemma 17.** *For any $w \in \Sigma^*$, $q^1, r^{(1)} \in Q_1$ and $q^2, r^2 \in Q_2$, the following are equivalent:*

**(i)** $((q^{(1)}, q^{(2)}), m_1, \ldots, m_{d_1+d_2}) \xrightarrow{w}_V ((r^{(1)}, r^{(2)}), m_1', \ldots, m_{d_1+d_2}')$

**(ii)** $(q^{(1)}, m_1, \ldots, m_{d_1}) \xrightarrow{w}_{V_1} (r^{(1)}, m_1', \ldots, m_{d_1}')$ *and* $(q^{(2)}, m_{d_1+1}, m_{d_1+1} \ldots, m_{d_1+d_2}) \xrightarrow{w}_{V_2}$ $(r^{(1)}, m_{d_1+1}', \ldots, m_{d_1+d_2}')$.

**Proof.** Let $w \in \Sigma^*$, we prove the statement by induction on $|w|$. The base case $w = \epsilon$ is immediate by the definition of $V$.

For the induction step, let $w = u \cdot a$ for some $a \in \Sigma$. The induction hypothesis yields

$$\begin{aligned}
&((q^{(1)}, q^{(2)}), m_1, \ldots, m_{d_1+d+2}) \xrightarrow{u}_V ((s^{(1)}, s^2), m''_1, \ldots, m''_{d_1+d_2}) \\
\iff &(q^{(1)}, m_1, \ldots, m_{d_1}) \xrightarrow{u}_{V_1} (s^{(1)}, m''_1, \ldots, m''_{d_1}) \text{ and} \\
&(q^{(2)}, m_{d_1+1}, \ldots, m_{d_1+d_2}) \xrightarrow{u}_{V_2} (s^{(2)}, m''_{d_1+1}, \ldots, m''_{d_1+d_2}) .
\end{aligned}$$

Again, by definition of $V$ we furthermore have

$$\begin{aligned}
&((s^{(1)}, s^{(2)}), m''_1, \ldots, m''_{d_1+d+2}) \xrightarrow{a}_V ((r^{(1)}, r^2), m'_1, \ldots, m''_{d_1+d_2}) \\
\iff &(s^{(1)}, m_1, \ldots, m_{d_1}) \xrightarrow{a}_{V_1} (r^{(1)}, m''_1, \ldots, m''_{d_1}) \text{ and} \\
&(s^{(2)}, m_{d_1+1}, \ldots, m_{d_1+d_2}) \xrightarrow{a}_{V_2} (r^{(2)}, m''_{d_1+1}, \ldots, m''_{d_1+d_2}) .
\end{aligned}$$

This concludes the proof of the statement. ◀

▶ **Corollary 18.** *Let $V_1, V_2$ be LAVASS. Then $L(V_1) \cap L(V_2)) = L(V_1 \cap V_2)$.*

**Proof.** For any $w \in \Sigma^*$, by Lemma 17, we have:

$$\begin{aligned}
&w \in L(V_1 \cap V_2) \\
\iff &(q'_0, 0, \ldots, 0) \xrightarrow{w}_V ((r_1, r_2), m_1, \ldots, m_{d_1+d_2}) \\
&\quad \text{for some } (r_1, r_2) \in F', (m_1, \ldots, m_{d_1+d_2}) \in S'_f \\
\iff &(q_0^{(1)}, 0, \ldots, 0) \xrightarrow{w}_{V_1} (r_1, m_1, \ldots, m_{d_1}) \text{ for some } r_1 \in F_1, (m_1, \ldots, m_{d_1}) \in S_f^{(1)}, \text{ and} \\
&(q_0^{(2)}, 0, \ldots, 0) \xrightarrow{w}_{V_2} (r_2, m_{d_1+1}, \ldots, m_{d_1+d_2}) \\
&\quad \text{for some } r_2 \in F_2\, (m_{d_1+1}, \ldots, m_{d_1+d_2}) \in S_f^{(2)} \\
\iff &w \in L(V_1) \text{ and } w \in L(V_2) .
\end{aligned}$$ ◀

From the construction, it is clear that for the size of the LAVASS for $L(V_1) \cap L(V_2)$, we have:

- $|Q| = |Q_1| \cdot |Q_2|$,
- $|\Delta| \leq |\Delta_1| \cdot |\Delta_2|$,
- $|\lambda| = max(|\lambda_1|, |\lambda_2|)$, and
- $d = d_1 + d_2$.

## A.2 Proof of Theorem 1

We show how Theorem 1 follows from Proposition 9. Given a formula $\Phi$ of generalised Semënov arithmetic, we can in space $2^{O(|\Phi|)}$ construct the disjunctive normal form of $\Phi$. Every disjunct can be assumed to be of the form

$$A \cdot \boldsymbol{x} = \boldsymbol{b} \wedge \bigwedge_{i \in I} x_i = 2^{y_i} \wedge \bigwedge_{j \in J} R_j(\boldsymbol{x}),$$

where the $R_j$ are predicates over regular languages. By Lemma 8, there is a restricted LAVASS for $\Phi$ of dimension $2|I|$ with a number of states bounded by $(\|A\|_{1,\infty} + \|\boldsymbol{b}\|_\infty + 2)^{O(m+|I|)} = 2^{p(|\Phi|)}$ for some polynomial $p$ and whose language represents the set of solutions to $A \cdot \boldsymbol{x} = \boldsymbol{b} \wedge \bigwedge_{i \in I} x_i = 2^{y_i}$. Intersecting with the DFA for the $R_j$ results in a restricted LAVASS $V$ with $2|I| = O(|\Phi|)$ counters such that $|V| = 2^{p(|\Phi|)}$ for some polynomial $p$. By Proposition 9, it follows that emptiness of $V$ is decidable in NSPACE$(2^{p(|\Phi|)} \cdot 2^{2|I|})$. We conclude the argument by recalling that NEXPSPACE=EXPSPACE by Savitch's theorem.

## A.3 Witnessing certificates imply language non-emptiness

To formally prove Proposition 12, let $(R, X, Y, L)$ be a witnessing certificate, and let $\pi(R)$ be the run in the configuration graph of $V$ induced by $R$. Let $a \leq d$ be maximal such that $\ell_a \neq \top$. We now define a sequence of runs $\pi_0, \dots, \pi_a$ such that the following invariant holds. In the final configuration of $\pi_i$,

**(i)** $m_j \leq n_j$ and $m_j \equiv n_j \bmod \ell_j$ for the $j$-th counter pair, $1 \leq j \leq a - i$; and
**(ii)** $m_j = n_j$ for the $j$-th counter pair, $a - i < j \leq d$.

It is clear that $\pi_a$ then witnesses $L(V) \neq \emptyset$. We proceed by induction on $i$.

**Base case $i = 0$.** Let $\pi_0 = \pi(R)$. Since $R$ is a witnessing certificate, $val(\pi(R), x_a) \leq val(\pi(R), y_a)$, and hence $m_a \leq n_a$ in the last configuration of $\pi_0$. Moreover, $R$ respects the set of induced $y$-constraints. Hence $n_{a-1} - n_a \geq \delta_{a-1}$, where $\delta_{a-1}$ is the length of the path from $R[X(a-1)]$ to $R[X(a)]$. Hence $n_{a-1} - n_a \geq m_{a-1} - m_a$ and thus $m_{a-1} \leq n_{a-1}$. Iterating this argument for the remaining counters, we get that (i) of the invariant is fulfilled for $\pi_0$; (ii) trivially holds since $R$ ends in an accepting abstract configuration.

**Induction step $i > 0$.** Let $\pi_{i-1}$ be the path that exists by the induction hypothesis. If $m_{a-i} = n_{a-i}$ in the last configuration of $\pi_{i-1}$ then we are done and take $\pi_i = \pi_{i-1}$; otherwise $m_{a-i} < n_{a-i}$ and $m_{a-i} \equiv n_{a-i} \bmod \ell_{a-i}$. Hence, there is some $k \in \mathbb{N}$ such that $n_{a-i} = k \cdot \ell_{a-i}$. Since $\ell_i \neq \top$, let $\beta := \alpha_{L(a-i)} \xrightarrow{t_1} \alpha_2 \xrightarrow{t_2} \cdots \alpha_{\ell_i - 1} \xrightarrow{t_{\ell_i}} \alpha_{L(a-i)}$ be the simple $\alpha$-loop at position $L(a-i)$ that is guaranteed to exist since $R$ is a witnessing certificate. We insert the transitions of $\beta^k$ and the induced updated configurations into $\pi_{i-1}$ at position $L(a-i)$. Notice that $L(a-i) < X(a-i+1)$. Otherwise, by the definition of the induced $y$-constraints, $y_{a-i} - y_{a-i+1} = \delta_{a-i}$ is in the set of induced $y$-constraints, where $\delta_i = X(a-i+1) - X(a-i)$. Since the last abstract configuration of $R$ respects the set of $y$-constraints, it must be the case that in the last configuration of $\pi_{i-1}$, $n_{a-i} - n_{a-i+1} = \delta_{a-i}$ and $m_{a-i} - m_{a-i+1} = \delta_{a-i}$, so $m_{a-i} = n_{a-i}$, because after the position $X(a-i+1) - 1$ in $R$ and thus $\pi_{i-1}$, the counters $x_{a-i}, x_{a-i+1}$ get incremented simultaneously. This contradicts our assumption that $m_{a-i} \neq n_{a-i}$. Thus, the counters $x_{a-i+1}, y_{a-i+1}, \dots, x_d, y_d$ remain unchanged by the insertion of $\beta^k$, so (ii) and consequently (i) continues to hold in the last configuration of $\pi_i$ for those counters. Moreover, due to the ordering conditions imposed on witnessing certificates, the value of $y_{a-i}$ does not change either, and hence $m_{a-i} = n_{a-i}$ in the last configuration of $\pi_i$. Since $\beta$ is a loop in the abstract configuration space, we have $m_j \equiv n_j \bmod \ell_j$ for all $1 \leq j < a - i$ and the values of $u_j$, for all $1 \leq j < a$ are preserved.

## A.4 From string constraints to Semënov arithmetic

Again, we treat $T_{\mathrm{REln}}$ as a relational structure. Without loss of generality, we may assume that atomic formulas of $T_{\mathrm{REln}}$ are one of the following:

- $R(s)$ for some string variable $s$ and a regular language $R$;
- $s = t$ for some string variables $s$ and $t$;
- $len(s, x)$ or $sn(s, x)$ for some string variable $s$ and integer variable $x$; or
- $\boldsymbol{a} \cdot \boldsymbol{x} \geq b$ for a vector of integer variables $\boldsymbol{x}$.

The size of a formula of $T_{\mathrm{REln}}$ is defined in the standard way as the number of symbols required to write it down, assuming binary encoding of numbers, and where the size of some $R$ is the size of the smallest DFA accepting $R$. Furthermore, in a quantifier-free formula $\varphi$ of $T_{\mathrm{REln}}$, we may without loss of generality assume that all atomic formulas occur positive, except for atomic formulas $s = t$.

We now describe the reduction to existential Semënov arithmetic. As explained, the idea underlying our proof is that we map a string $s$ to the number $[\![1s]\!]$. Given a quantifier-free formula $\varphi$ of $T_{\text{REln}}$, we define by structural induction on $\varphi$ a function $\sigma$ that maps $\varphi$ to an equi-satisfiable formula of generalised Semënov arithmetic:

- Case $\varphi \equiv R(s)$: $\sigma(\varphi) := (0^*1R)(s)$;
- Case $\varphi \equiv s = t$ or $\varphi \equiv \neg(s = t)$: $\sigma(\varphi) := s = t$ or $\sigma(\varphi) := \neg s = t$, respectively;
- Case $\varphi \equiv \mathit{sn}(s, x)$: $\sigma(\varphi) := \exists y.\, 2^y \leq s \wedge s < 2^{y+1} \wedge x = s - 2^y$;
- Case $\varphi \equiv \mathit{len}(s, x)$: $\sigma(\varphi) := 2^x \leq s \wedge s < 2^{x+1}$;
- Case $\varphi \equiv \boldsymbol{a} \cdot \boldsymbol{x} \geq b$: $\sigma(\varphi) := \boldsymbol{a} \cdot \boldsymbol{x} \geq b$; and
- Case $\varphi \equiv \varphi_1 \sim \varphi_2$, $\sim\, \in \{\wedge, \vee\}$: $\sigma(\varphi) := \sigma(\varphi_1) \sim \sigma(\varphi_2)$.

▶ **Lemma 19.** *Let $\varphi$ be a quantifier-free formula of $T_{\text{REln}}$ and $S$ be the set of string variables occurring in $S$. Then $\varphi$ is satisfiable if and only if $\sigma(\varphi) \wedge \bigwedge_{s\in S} s > 0$ is satisfiable.*

**Proof.** Observe that the variables occurring in $\varphi$ are the same variables as those occurring in $\sigma(\varphi)$. Let $S$ be the set of string variables in $\varphi$ and $X$ be the set of integer-valued variables in $\varphi$. Given an assignment $\mathcal{I}_S\colon S \to \{0,1\}^*$, we define $\tilde{\mathcal{I}}_S := S \to \mathbb{N}$ such that $\tilde{\mathcal{I}}_S(s) := [\![1\mathcal{I}_S(s)]\!]$. Subsequently, denote by $\mathcal{I}_X : X \to \mathbb{N}$ an assignment to the integer-valued variables. We show by structural induction on $\varphi$ that $(\mathcal{I}_S, \mathcal{I}_x) \models \varphi$ if and only if $(\tilde{\mathcal{I}}_S, \mathcal{I}_X) \models \sigma(\varphi) \wedge \bigwedge_{s\in S} s > 0$:

- Case $\varphi \equiv R(s)$: Let $\mathcal{I}_S(s) = b_{n-1} \cdots b_0$, we have $\mathcal{I}_S(s) \in R$ if and only if $2^n + \sum_{i=0}^{n-1} 2^i b_i \in [\![0^*1R]\!]$, noting that $2^n + \sum_{i=0}^{n-1} 2^i b_i = [\![1b_{n-1} \cdots b_0]\!] = \tilde{\mathcal{I}}_S(s)$.
- Case $\varphi \equiv \mathit{sn}(s, x)$: Let $\mathcal{I}_S(s) = b_{n-1} \cdots b_0$ and $\mathcal{I}_X(x) = m$. We have that $m = \sum_{i=0}^{n-1} 2^i b_i$ if and only if $m = \tilde{\mathcal{I}}_S(s) - 2^n$ if and only if $(\tilde{\mathcal{I}}_S, \mathcal{I}_X) \models \sigma(\varphi) \wedge \bigwedge_{s\in S} s > 0$.
- Case $\varphi \equiv \mathit{len}(s, x)$: Let $\mathcal{I}_S(s) = b_{n-1} \cdots b_0$ and $\mathcal{I}_X(x) = m$. We have that $m = n$ if and only if $2^m \leq [\![1b_{n-1} \cdots b_0]\!] < 2^{m+1}$ if and only if $(\tilde{\mathcal{I}}_S, \mathcal{I}_X) \models \sigma(\varphi) \wedge \bigwedge_{s\in S} s > 0$.

The remaining cases follow obviously. ◀