Algebraic Closure of Matrix Sets Recognized by 1-VASS

Rida Ait El Manssour
University of Oxford, UK

Mahsa Naraghi
Université Paris Cité, IRIF, France

Mahsa Shirmohammadi
CNRS, IRIF, France

James Worrell
University of Oxford, UK

Abstract

It is known how to compute the Zariski closure of a finitely generated monoid of matrices and, more generally, of a set of matrices specified by a regular language. This result was recently used to give a procedure to compute all polynomial invariants of a given affine program. Decidability of the more general problem of computing all polynomial invariants of affine programs with recursive procedure calls remains open. Mathematically speaking, the core challenge is to compute the Zariski closure of a set of matrices defined by a context-free language. In this paper, we approach the problem from two sides: Towards decidability, we give a procedure to compute the Zariski closure of sets of matrices given by one-counter languages (that is, languages accepted by one-dimensional vector addition systems with states and zero tests), a proper subclass of context-free languages. On the other side, we show that the problem becomes undecidable for indexed languages, a natural extension of context-free languages corresponding to nested pushdown automata. One of our main technical tools is a novel adaptation of Simon's factorization forests to infinite monoids of matrices.

1 Introduction

1.1 Context

Finitely generated matrix monoids are not recursive in general, and many of the associated computational problems are undecidable. In particular, the Mortality Problem (determine whether a given finitely generated matrix monoid contains the zero matrix) is undecidable already in dimension 3 [35]. On the other hand, it is known how to decide certain global properties of matrix semigroups. For example, one can determine whether a finitely generated sub-monoid of the monoid $M_d(\mathbb{Q})$ of $d \times d$ rational matrices is finite [6, 23, 29]. More generally, there is a procedure to compute the Zariski closure of finitely generated matrix monoids [20]. The output of this algorithm is a finite collection of polynomials that capture all the algebraic relationships between the different entries of each matrix in the monoid (such as the property that the determinant of every matrix equals one). The set of common zeros of the latter collection of polynomials is the Zariski closure of the monoid.

The algorithm for computing the Zariski closure of a matrix monoid admits a straightforward generalization that, given a regular language $L \subseteq \Sigma^*$ and a monoid morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$, allows computing the Zariski closure of the set $\varphi(L)$ in $M_d(\mathbb{Q})$. Such a procedure was used in [20] to resolve a problem posed by Müller-Olm and Seidl [32] on the computability of polynomial invariants in program analysis. A special case of this result, concerning the linear Zariski closure, was recently used by Bell and Smertnig [2] to resolve a longstanding open problem concerning ambiguity in formal languages and automata. The procedure of [20] generalizes and builds on a procedure of [13] which solves the version of the problem for matrix groups and which was applied to study language

emptiness for quantum automata. There is a wealth of related work on the automatic synthesis of program invariants and procedure summaries; a select sample includes [9, 21, 24, 36, 30].

In this paper we are interested in computing the Zariski closure of sets of matrices $\varphi(L)$ for languages L beyond regular. One of our main motivations is an application to interprocedural program analysis, where the problem of computing all polynomial invariants for a collection of affine programs that feature recursive procedure calls (as studied in [7, 17, 16, 33, 31], among many other sources) can be understood in terms of computing the Zariski closure of $\varphi(L)$ for L a context-free language. In the presence of recursive procedure calls, regular languages no longer suffice to model all paths through a control-flow graph that models a program. Figure 1 shows two recursive affine programs, in the sense of [33], in which the the set of execution paths is defined by a one-counter language. Affine programs feature integer-valued variables that can be updated by linear assignments. In the interests of decidability, the boolean conditionals in loop guards and control-flow statements are abstracted to wildcards ($*$) that non-deterministically evaluate to true or false. At present it is open whether one can compute all interprocedural polynomial invariants, not just those of an *a priori* bounded degree, in the setting of affine programs (see the discussion in [31, Section 8]).

```
procedure P()
  begin
    if (*) then
      x := Ax
      call P()
      x := Bx
    else
      skip
    endif
  end
```

```
procedure Q()
  begin
    if (*) then
      x := Ax
      call Q()
      x := Bx
      call Q()
    else
      skip
    endif
  end
```

Figure 1: Two recursive procedures in which all variables are global. The wildcard (*) evaluates non-deterministically either to true or false. An execution of the procedure P on the left can be summarized by the assignment $\boldsymbol{x} := B^n A^n \boldsymbol{x}$ for some $n \in \mathbb{N}$. An execution of the procedure Q on the right results in assignment $\boldsymbol{x} := M\boldsymbol{x}$ where the matrix M is a product of a string of A's and B's that is well-matched in the sense of the Dyck language.

1.2 Main Results

The main decidability results in the paper concern one-counter languages, which are a special case of context-free languages. In fact, our technical development uses the terminology and formalism of 1-VASS (one-dimensional vector addition systems with states). A 1-VASS is a non-deterministic finite automaton that is equipped with a single counter that takes values in the nonnegative integers and that can be incremented and decremented on transitions. Decrement transitions cannot be taken when the counter is zero. Acceptance is either by control state, in which case we speak of the *coverability language*, or by control state and counter value, in which case we speak of the *reachability language*. Perhaps the best known example of a 1-VASS reachability language is the Dyck language of well-parenthesized words on a binary alphabet. A one-counter automaton is a 1-VASS with additional transitions, called *zero tests*, that can only be taken when the counter is zero. As we will see, it is straightforward to extend our decidability results to one-counter automata.

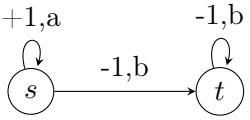
Let $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ be a monoid morphism. Given a 1-VASS coverability or reachability language $L \subseteq \Sigma^*$, we are interested in computing the set of all polynomials in $\mathbb{Q}[\{x_{ij}\}_{1\leq i,j\leq d}]$ that vanish on the set $\varphi(L) = \{\varphi(w) : w \in L\}$ of matrices. In general, this set of polynomials forms an ideal in the polynomial ring $\mathbb{Q}[\{x_{ij}\}_{1\leq i,j\leq d}]$ and the goal is to compute a finite basis of this ideal (Such a finite basis necessarily exists by the Hilbert basis theorem). The set of common zeros of these polynomials is the *Zariski closure* of $\varphi(L)$. We regard this as a subset of $M_d(\overline{\mathbb{Q}})$, where $\overline{\mathbb{Q}}$ is the set of algebraic numbers (that is, complex numbers that are roots of univariate polynomials in $\mathbb{Q}[x]$). The Zariski closure is thus an over-approximation of the original set, capturing all polynomial relations satisfied by the entries of its matrices.

The first main contribution of the paper is a procedure that takes as input a 1-VASS coverability language L on alphabet Σ and a morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$, and outputs a finite collection of polynomials that generates the vanishing ideal of $\varphi(L)$. We also present a more general procedure for computing the vanishing ideal of $\varphi(L)$ for L a 1-VASS reachability language. We complement the above two results by showing undecidability of computing the Zariski closure of $\varphi(L)$, where L is an indexed language. Recall that indexed grammars [1] extend context-free grammars by allowing stack-like memory on nonterminals. Indexed grammars correspond to second-order pushdown automata, in which the main stack can hold secondary stacks as element. A classic example of an indexed language that is not context-free is $\{a^n b^n c^n : n \in \mathbb{N}\}$.

1.3 Examples

The following examples illustrate the above results. The steps of the computations are shown in Section A. Recall that a 1-VASS is a non-deterministic finite automaton in which each transition is annotated by an integer weight. The *coverability language* is the set of all words arising from an accepting run of the automaton in which every prefix has non-negative weight. The *reachability language* is the set of all words arising from accepting runs of total weight zero in which every prefix has non-negative weight. In the examples below, the state s is initial and the state t is accepting.

Example 1. Consider the following 1-VASS with the reachability language $L_R = \{a^n b^n \mid n \in \mathbb{N}\}$, and the monoid morphism $\varphi : \{a, b\}^* \to M_2(\mathbb{Q})$ as defined below.



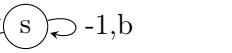
$$\varphi(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},\ \varphi(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

The Zariski closure $\overline{\varphi(L_R)} \subseteq M_2(\overline{\mathbb{Q}})$ is the zero set of the ideal $I_R \subseteq \mathbb{Q}[x_{11}, x_{12}, x_{21}, x_{22}]$ generated by the polynomial relations between entries of the matrices in $\varphi(L_R)$. As the image of the word $a^n b^n$ under φ is $\begin{pmatrix} n^2+1 & n \\ n & 1 \end{pmatrix}$, by simple algebraic manipulations one obtains that

$$\langle x_{11} - x_{12}x_{21} - 1,\ x_{12} - x_{21},\ x_{22} - 1\rangle.$$

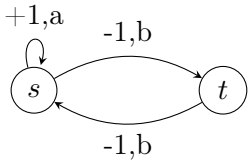
Similarly, the Zariski closure $\overline{\varphi(L_C)} \subseteq M_2(\overline{\mathbb{Q}})$ is the zero set of $\langle x_{11} - x_{12}x_{21} - 1,\ x_{22} - 1\rangle$. ◀

Example 2. A second simple example concerns the so-called Dyck language L, consisting of all words with same number of a's and b's such that every prefix has at least as many a's and b's. This is the reachability language of the following 1-VASS:



Since L is closed under concatenation, its image under the above morphism φ is closed under multiplication. Moreover, since $\varphi(a)$ and $\varphi(b)$ are invertible matrices, $\varphi(L)$ is a sub-semigroup of $GL_2(\mathbb{Q})$. Due to Fact 9, the Zariski closure of $\varphi(L)$ inside $GL_2(\overline{\mathbb{Q}})$ is a linear algebraic subgroup. Since both $\varphi(a)$ and $\varphi(b)$ have determinant one, all elements $\varphi(L)$ also have determinant one. In this case, the ideal of all polynomials relations satisfied by $\varphi(L)$ is the principal ideal generated by $x_{11}x_{22} - x_{12}x_{21} - 1$, expressing that the determinant is one. In other words, the Zariski closure is the special linear group $SL_2(\overline{\mathbb{Q}})$ of all 2×2 matrices with algebraic entries and determinant one. ◀

Example 3. Consider the following 1-VASS, whose coverability language L_C is a subset of the regular language $a^*(a^*b^2)^*b$. For every word $w \in L_C$, every prefix of w contains at least as many occurrences of a as of b.



The reachability language L_R of this 1-VASS is the subset of L_C determined by the condition that the total number of a's and b's in a word be equal. For the morphism φ_1 and ideal I shown below, the respective closures of $\varphi_1(L_C)$ and $\varphi_1(L_R)$ coincide and are the zero set of I:

$$\varphi_1(a) = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix},\ \varphi_1(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \qquad\qquad I := \langle x_{12},\, x_{11}^2 - x_{22} \rangle.$$

As a final example consider the set of matrices $\varphi_2(L_C)$ arising from the morphism φ_2, shown below:

$$\varphi_2(a) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix},\ \varphi_2(b) = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

The closure of $\varphi_2(L_C)$ is the zero set of $I_C := \langle x_{11}-1, x_{22}-1, x_{33}-1, x_{21}, x_{31}, x_{32} \rangle$. The defining ideal of the closure of $\varphi_2(L_R)$ is $\langle I_C, x_{12} \rangle$, where the additional equation $x_{12} = 0$ reflects the condition that total number of a's and b's in every word $w \in L_R$ is equal. ◀

1.4 Discussion

The existence of a procedure for computing the Zariski closure of $\varphi(L)$, given a context-free language L and a morphism φ, remains open. Extending beyond interprocedural program analysis to models of concurrency and parameterized systems, another natural next step is to consider the case where L is the coverability or reachability language of a VASS in dimension greater than one. We note that Presburger invariants for VASS reachability sets were studied in [25], leading to a new algorithm for solving VASS reachability [26]. (The VASS reachability problem has attracted renewed interest following the resolution of a long-standing question concerning its complexity [10, 28, 11, 27].) Comparing the result of [25] to the generalization of Reach Closure to general VASS, we note on the one hand that the reachable set of configurations of a given VASS can easily be expressed as a morphic image of its reachability language but, on the other hand, Presburger invariants involve linear inequalities whereas Reach Closure concerns polynomial equations.

Our work also has similarities to a line of research on regular separability of VASS languages [10, 12]. Here, the question is to determine, given two VASS, whether there is a regular language that contains the language of one VASS and is disjoint from the language of the other. For comparison, our solution to Cover Closure involves computing a regular language that contains a given 1-VASS language and has the same Zariski closure.

2 Summary and Overview

This section contains statements of the main results in the paper and a high-level overview of the technical approach. The detailed development is given in the remainder of the paper.

2.1 Stateless Setting

To study the problems of computing the Zariski closure of (the morphic images) of 1-VASS coverability and reachability languages in an abstract setting, we introduce two families of languages. Given a morphism $\omega : \Sigma^* \to \mathbb{Z}$ we define the language $L_C(\omega)$, comprising those words whose prefixes all have nonnegative ω-value, and the language $L_R(\omega)$, comprising those words in $L_C(\omega)$ that also have value zero. Formally we have:

$$\begin{aligned} L_C(\omega) &:= \{w \in \Sigma^* : \forall u \preceq w \, (\omega(u) \geq 0)\} \\ L_R(\omega) &:= \{w \in \Sigma^* : \omega(w) = 0 \wedge \forall u \preceq w \, (\omega(u) \geq 0)\}. \end{aligned}$$

where $\preceq$ denotes the prefix relation on words. The core of our technical development addresses the following two decision problems: given morphisms $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$, the Cover Closure problem asks to compute the Zariski closure of $\varphi(L_C(\omega))$, while the Reach Closure problem asks to compute the Zariski closure of $\varphi(L_R(\omega))$. We then have:

Proposition 1. The problem of computing the Zariski closure of a 1-VASS coverability language is interreducible with Cover Closure. The problem of computing the Zariski closure of a 1-VASS reachability language is interreducible with Reach Closure.

In the proof of Proposition 1, given in Section B, the arguments for coverability and reachability are essentially the same; here we sketch the former. In brief, the problem of computing the closure of a 1-VASS coverability language can be reduced to the special case of 1-VASS with a single control state by blowing up the dimension of the matrices. Second, the version of the problem for single-state 1-VASS is equivalent to Cover Closure since every language $L_C(\omega)$ is the coverability language of a single-state 1-VASS and, conversely, every such coverability language is the homomorphic image of a language of the form $L_C(\omega)$. As noted in Section B, a corollary of Proposition 1 is that in showing decidability of Cover Closure and Reach Closure we may assume without loss of generality that every symbol $\sigma \in \Sigma$ has weight $\omega(\sigma) \in \{-1, 0, +1\}$. We make this assumption henceforth.

A 1-VASS $\mathcal{V}$ with *zero tests* features transitions that can only be taken when the counter is zero. The reachability language of $\mathcal{V}$ can be effectively represented by a regular expression whose atoms are reachability languages of 1-VASS without zero tests (decompose each accepting run of $\mathcal{V}$ into segments between consecutive zero-tests). This observation allows us to reduce the problem of computing $\overline{\varphi(L(\mathcal{V}))}$ corresponding problem for 1-VASS without zero tests which, by Proposition 1, is equivalent to Reach Closure.

2.2 Invertible Matrices

In the special case that the morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ takes values in the group of invertible matrices, we can use a Noetherian property for ascending chains of linearly algebraic groups, due to [34], to show that Cover Closure and Reach Closure are both computable. To explain this, we first note that by Fact 9, the condition on φ entails that $\overline{\varphi(L_C(\omega))}$ and $\overline{\varphi(L_R(\omega))}$ are both linear algebraic groups (not just monoids). Now [34, Theorem 11] states that every strictly increasing chain $G_1 \subsetneq G_2 \subsetneq \cdots \subsetneq G_\ell \subseteq \ldots$ of Zariski-closed groups of invertible matrices in $M_d(\overline{\mathbb{Q}})$

is finite, provided that each G_i are topologically generated by rational matrices. (Here we say that G is *topologically generated* by a set S if $G = \overline{\langle S \rangle}$.)

We can use this Noetherian property to compute $\overline{\varphi(L_C(\omega))}$ and $\overline{\varphi(L_R(\omega)}$ as follows: First, we partition the alphabet Σ into subsets Σ_0, Σ_1, and Σ_{-1}, consisting of letters with weights 0, 1, and -1, respectively. There are then two cases:

1. Cover Closure: We define G_0 to be the group topologically generated by $\varphi(\Sigma_0 \cup \Sigma_1)$, and for all $i \in \mathbb{N}$, we write
$$G_{i+1} = \overline{\langle \varphi(\Sigma_1)\, G_i\, \varphi(\Sigma_{-1})\,,\, G_i \rangle}\,.$$

2. Reach Closure: We define G_0 to be the group topologically generated by $\varphi(\Sigma_0)$ and for all $i \in \mathbb{N}$, similar to the previous case, take G_{i+1} to be topologically generated by
$$G_{i+1} = \overline{\langle \varphi(\Sigma_1)\, G_i\, \varphi(\Sigma_{-1})\,,\, G_i \rangle}\,. \tag{1}$$

We will show that once these chains stabilize, the resulting group is precisely the Zariski closure of the monoid of interest. Here we give the proof in the case of Reach Closure, the proof for Cover Closure follows similarly.

Proposition 2. Consider the chain of groups G_i constructed in Equation (1), and let m be the smallest integer such that $G_m = G_{m+1}$. Then $\overline{\varphi(L_R(\omega))} = G_m$.

Proof. First, let us prove that if there are two consecutive equivalent elements in the chain then the chain would stabilize. Let m be such that $G_m = G_{m+1}$. Then we have that
$$G_{m+2} = \overline{\langle \varphi(\Sigma_1)\, G_{m+1}\, \varphi(\Sigma_{-1}), G_{m+1} \rangle} = \overline{\langle \varphi(\Sigma_1)\, G_m\, \varphi(\Sigma_{-1}), G_m \rangle} = G_m\,.$$
Hence, $G_{m+k} = G_m$ for every $k \in \mathbb{N}$.

Next, we define a sequence of languages in recursive way as follows:
$$L_0 = \Sigma_0 \qquad \text{and} \qquad L_{i+1} = (\Sigma_1 L_i \Sigma_{-1} \cup L_i)^*\,.$$
By an induction on i, it follows that $G_i = \overline{\varphi(L_i)}$. Moreover, we have $L_R(\omega) = \bigcup_{i\in\mathbb{N}} L_i$. Thus,
$$\overline{\varphi(L_R(\omega))} = \overline{\bigcup_{i\in\mathbb{N}} \overline{\varphi(L_i)}} = G_m.$$
□

The given construction shows that the problems Cover Closure and Reach Closure are decidable in the case of invertible matrices. However, this construction breaks down for non-invertible matrices, as shown by the following example.

Example 4. Let $\Sigma = \{\alpha, \beta, \sigma_1, \sigma_2\}$. Define $\varphi : \Sigma^* \to M_2(\mathbb{Q})$ and by
$$\varphi(\sigma_1) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \qquad \varphi(\sigma_2) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad \varphi(\alpha) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \qquad \varphi(\beta) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix},$$
and $\omega : \Sigma^* \to \mathbb{Z}$ by
$$\omega(\sigma_1) = \omega(\sigma_2) = 0 \qquad\qquad \omega(\alpha) = -\omega(\beta) = 1\,.$$
Let S_0 be the monoid topologically generated by $\varphi(\sigma_1)$ and $\varphi(\sigma_2)$. We write $S_0 := \overline{\langle \varphi(\sigma_1), \varphi(\sigma_2) \rangle}$, and observe that S_0 is a finite set containing the two generators and identity matrix. Consider

the chain $(S_i)_{i\geq 0}$ defined inductively by $S_{i+1} = \langle \varphi(\alpha)\, S_i\, \varphi(\beta)\,,\, S_i\rangle$, as described above for Reach Closure. *But this chain does not stabilize.* Indeed, we claim that for each $i \in \mathbb{N}$,

$$S_i = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 2^j \\ 0 & 0 \end{pmatrix} ; 0 \leq j \leq i \right\}.$$

The base case ($i = 0$) holds by definition of S_0. For the inductive step, observe that $\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 2^j \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 2^{j+1} \\ 0 & 0 \end{pmatrix}$. Thus, each iteration introduces a new matrix with an entry 2^{j+1} in the upper-right corner, and so the chain $(S_i)_{i\geq 0}$ grows strictly at each step, never stabilizing. ◀

Solving the Cover Closure and Reach Closure in the general case—without the invertibility assumption—appears to be more challenging. The following section introduces the key technical tool that underlies our approach in the general setting.

2.3 Factorization Trees for Matrix Monoids

We say that a matrix $M \in M_d(\mathbb{Q})$ is *stable* if M and all its powers have the same rank. An equivalent condition is that $\mathrm{im}(M) \cap \ker(M) = \{0\}$ where $\mathrm{im}(M)$ and $\ker(M)$ respectively denote the image and kernel of the linear transformation $x \mapsto Mx$. Using the notion of stable matrix we adapt the notion of factorization tree from finite monoids to the infinite monoid $M_d(\mathbb{Q})$. The original notion of factorization tree was introduced by Imre Simon [37] in his resolution of the star-height problem for regular languages (see also [3]). In our variant, the role of idempotent elements in the classical version is replaced by stable matrices.

We define a factorization tree in $M_d(\mathbb{Q})$ to be a finite tree in which each node is labeled by an element of $M_d(\mathbb{Q})$, subject to two conditions:

- Product condition: If a node has children with labels $M_1, \ldots, M_k$ (read from left to right), then the label of the node is the matrix product $M_1 \cdots M_k$.
- Stability condition: If a node has more than two children, then each of its children must be a stable matrix.

The *height* of a factorization tree is defined as the maximum distance from the root to any leaf. The *yield* of the tree is the sequence of leaf labels read from left to right, which is a sequence of matrices in $M_d(\mathbb{Q})$. Using manipulations in multilinear algebra we prove that every sequence of matrices in $M_d(\mathbb{Q})$, no matter its length, can be realized as the yield of factorization tree of height bounded by the explicit function $\theta(d)$ of the dimension d.

Proposition 3. Every sequence of matrices in $M_d(\mathbb{Q})$ is the yield of some factorization tree of height at most $\theta(d) := d(d+3)$.

We use Proposition 3 to prove the following result, which shows that all words of a sufficiently large positive weight contain a stable factor with positive weight and, dually, all words of a sufficiently large negative weight contain a stable factor with negative weight.

Proposition 4. There exists a threshold $\eta(d) := 2^{d(d+3)} + 1$ such that for all $w \in \Sigma^*$:

(i) If $\omega(w) = \eta(d)$, then there exists a factor u of w such that $\omega(u) > 0$ and $\varphi(u)$ is stable.

(ii) If $\omega(w) = -\eta(d)$, then there exists a factor u of w such that $\omega(u) < 0$ and $\varphi(u)$ is stable.

2.4 Approximation by Regular Languages

We use our results on factorization trees to reduce the Cover Closure and Reach Closure problems to the problem of computing the Zariski closure of a set of matrices $\varphi(L)$ for a regular language L and morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$, for which there is a known algorithm [20]. This yields:

Theorem 5. The Cover Closure and Reach Closure problems are decidable.

Even though there is an easy reduction of Cover Closure to Reach Closure, it is instructive to give separate proofs of the decidability of Cover Closure.

At a high level, the proof strategy for showing decidability of Cover Closure is to compute an NFA $\mathcal{A}$ such that $L_C(\omega) \subseteq L(\mathcal{A})$ and $\overline{\varphi(L_C(\omega))} = \overline{\varphi(L(\mathcal{A}))}$. Automaton $\mathcal{A}$ keeps a count of the weight of the input word up to a certain threshold. If the weight of the input word ever exceeds the threshold the automaton moves to a sink state from which it accepts all words. This means that up to the threshold, the automaton can check that the weight of the word read so far is not negative, but thereafter there are no guarantees on the weight of the prefixes.

By construction, $L(\mathcal{A})$ is a superset of $L_C(\omega)$. In the other direction, using Proposition 4, we are able to show that every word $w \in L(\mathcal{A}) \setminus L_C(\omega)$ admits a factorization $w = xuy$ such that $xu \in L_C(\omega)$, $\varphi(u)$ is stable, and $\omega(u) > 0$. It follows that for all sufficiently large n we have $xu^{n+1}y \in L_C(\omega)$ since by pumping the positive-weight segment we assure that all prefixes have non-negative weight. By the fact that $\varphi(u)$ is stable, the Zariski-closed monoid $\overline{\{\varphi(u^n) : n \in \mathbb{N}\}}$ is in fact a group. In particular, this set contains a matrix I_u such that $\varphi(u)\, I_u = I_u\, \varphi(u) = \varphi(u)$. We conclude that $w \in \overline{\{xu^{n+1}y : n \in \mathbb{N}\}}$ and hence $w \in \overline{L_C(\omega)}$.

The following simple but instructive example shows that for reachability languages there may not be regular superset that has the same Zariski closure.

Example 5. Consider the alphabet $\Sigma = \{a, b\}$ and morphisms $\varphi : \Sigma^* \to M_1(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$ given by $\varphi(a) = 2, \varphi(b) = 1/2$ and $\omega(a) = 1, \omega(b) = -1$. The reachability language $L_R(\omega)$ comprises all words in $\{a, b\}^*$ with the same number of a's as b's. Thus $\varphi(L_R(\omega))$ is the singleton $\{1\}$.

By the Myhill-Nerode Theorem, for any regular language L' such that $L_R(\omega) \subseteq L'$, there exists an infinite sequence $m_1 < m_2 < m_3 < \cdots$ yielding the same left derivatives $(a^{m_i})^{-1}L'$. By this and the fact that $a^{m_i}b^{m_i} \in L'$, all words $a^{m_i}b^{m_j} \in L'$ for all $i, j \geq 1$. It follows that $\overline{\varphi(L')} = \overline{\mathbb{Q}}$. Hence there is no regular language that includes $L_N(\omega)$ and has the same Zariski closure.

Note, however, that for any morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ there is a simple method to compute $\overline{\varphi(L_R(\omega))}$. This set is the closure of the image of $\overline{\langle M \rangle}$ under the map $\Psi : M_{2d}(\overline{\mathbb{Q}}) \to M_d(\overline{\mathbb{Q}})$, where

$$M = \begin{pmatrix} \varphi(a) & 0 \\ 0 & \varphi(b) \end{pmatrix} \qquad \text{and} \qquad \Psi \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} = XY\,.$$

It thus suffices to compute the Zariski closure of the cyclic monoid $\langle M \rangle$, which is well understood (see, e.g., [13]). ◀

Our method to show decidability of Reach Closure is inspired by Example 5. Given a morphism $\omega : \Sigma^* \to \mathbb{Z}$, we compute a finite-state automaton $\mathcal{A}$ over an alphabet $\Gamma \subseteq (\Sigma^*)^4$ such that $\{\varphi(xywz) : (x, y, w, z) \in L(\mathcal{A})\}$ and $\varphi(L_R(\omega))$ have the same Zariski closure. As in the case of the Cover Closure problem, Proposition 4 is instrumental in the construction of $\mathcal{A}$. The automaton is based on the decomposition of words $w \in L_R(\omega)$ into three parts $w = w_1w_2w_3$ such that w_1 contains a factor u with $\varphi(u)$ stable and $\omega(u) > 0$ and w_2 contains a factor v with $\varphi(v)$ stable and $\omega(v) < 0$ as shown in 4. The details of the construction are much more involved than for the corresponding case of Cover Closure. In particular, instead of abstractly identifying all sufficiently large counter values, as in the former case, we treat counter values exactly by encoding them in the input word to the automaton $\mathcal{A}$.

2.5 Indexed Languages

We show that for sets of matrices defined by extensions of context-free languages, it is not possible to compute the Zariski closure.

Theorem 6. There is no algorithm that given an indexed grammar G over an alphabet Σ and a monoid morphism $\varphi : \Sigma^* \to \mathrm{M}_d(\mathbb{Q})$, computes the algebraic closure of $\varphi(L(G))$.

The proof of undecidability is by reduction from the boundedness problem for reset VASS. A reset VASS is a finite-state automaton equipped with multiple counters taking values in non-negative integers. The counters can be incremented, decremented, or reset to zero. A counter with value zero cannot be decremented, but there is no zero test (a transition that only be taken when a counter is zero). For this model, the boundedness problem ("is the set of reachable configurations finite?") is undecidable [15].

The essence of our reduction is to take a reset VASS and produce an indexed language $L \subseteq \Sigma^*$, each word of which encodes a run of the VASS. The encoding of a run is a complete list of all prefixes of the sequence of transitions in the run. We next construct a morphism φ with the property that $\overline{\varphi(L)}$ has dimension at most one if and only if the reachable set of the VASS is finite. The morphism φ maps each letter of Σ to a matrix that simulates the operation of incrementing and decrementing the VASS counters. The key challenge is to model the fact that a zero counter cannot be decremented. Our simulation is such that for any word $w \in \Sigma$ that represents an illegal run of the VASS (in which some counter takes a negative value) we have $\varphi(w) = \mathbf{0}$. Thus only legal computations yield non-zero elements of $\varphi(L)$. This is where the extra expressiveness of indexed grammars plays a key role, specifically the fact that our encoding of VASS runs as words is such that a word contains copy of each prefix of the run.

3 Factorization Trees in Matrix Monoids

Recall that $M \in M_d(\mathbb{Q})$ is *stable* if $\mathrm{rank}(M) = \mathrm{rank}(M^2)$. A *factorization tree* $\mathcal{T}$ in the monoid $M_d(\mathbb{Q})$ is a finite unranked ordered tree: a node can have arbitrarily many children and the children of each node are linearly ordered. The nodes of $\mathcal{T}$ are labeled by elements of $M_d(\mathbb{Q})$ such that the label of a non-leaf node equals the left-to-right product of the labels of its children. We require that the label M of a node with three or more children be stable. The *height* of $\mathcal{T}$ is the maximum distance from the root to a leaf. If $M_1, \ldots, M_m$ is a list of the labels of the leaf nodes, left-to-right, then we say that $\mathcal{T}$ is a factorization tree of the sequence $M_1, \ldots, M_m$. We refer to this sequence as the *yield* of the tree $\mathcal{T}$.

We call a sequence $M_1, \ldots, M_m \in M_d(\mathbb{Q})$ such that each matrix M_i and their product $M_1 \cdots M_m$ have rank r a *rank-r sequence*. A consecutive subsequence $M_i, M_{i+1} \ldots, M_j$, with $1 \leq i < j \leq m$, of the sequence $M_1, \ldots, M_m$ is called a *segment*.

Proposition 7. All rank-r sequences in $M_d(\mathbb{Q})$ are the yield of some factorization tree of height at most $d + 2$.

Proof. Recall that for a vector space V the *exterior algebra* $\Lambda(V)$ is a vector space that contains a copy of V and is equipped with a bilinear, associative, anti-symmetric operation $\wedge$. Below, we recall the key properties of $\Lambda(V)$ needed to follow the proof, but see Section C for mode details. If V has finite dimension d then $\Lambda(V)$ has dimension 2^d. The key properties of $\Lambda(V)$ for our purposes are that $v_1, \ldots, v_r \in V$ are linearly independent if and only if $v_1 \wedge \ldots \wedge v_r \neq 0$, and two such wedge products are nonzero scalar multiples of each other if and only if the corresponding sets of vectors

span the same r-dimensional subspace. This defines a map ι, which assigns to each r-dimensional subspace W of V a vector

$$\iota(W) := v_1 \wedge \ldots \wedge v_r \,,$$

where $\{v_1, \ldots, v_r\}$ is an arbitrarily chosen basis of the subspace [1]. Moreover, for subspaces W_1, W_2 of V, their intersection is trivial if and only if $\iota(W_1) \wedge \iota(W_2) \neq 0$.

The following claim is straightforward (see the proof in Section D):

Claim 7.1. Let $M_1, \ldots, M_m \in M_d(\mathbb{Q})$ be a rank-r sequence. Then $\mathrm{rk}(M_i \cdots M_k) = r$ for all $1 \leq i < k \leq m$. Furthermore $\mathrm{im}(M_{k+1}) \cap \ker(M_k) = \{0\}$ for $k \neq m$.

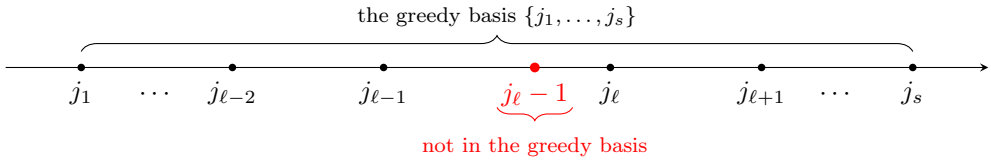
Fix a rank-r sequence $M_1, \ldots, M_m \in M_d(\mathbb{Q})$. We define the greedy basis of the subspace of $\Lambda(\mathbb{Q}^d)$ spanned by $\{\iota(\mathrm{im}\, M_i) \mid 1 \leq i < m\}$ to be the subsequence of indices $j_1 < \cdots < j_s$ such that $j_1 = 1$, and $\iota(\mathrm{im}(M_{j_p}))$ is not a linear combination of $\{\iota(\mathrm{im}\, M_i) \mid i < j_p\}$ for all $1 < p \leq s$. Note that

$$s \leq \dim \Lambda^r(\mathbb{Q}^d) \leq \dim \Lambda(\mathbb{Q}^d) = \sum_{r=0}^{d} \binom{d}{r} = 2^d \,. \tag{2}$$

Define $j_{s+1} := m + 1$.

Claim 7.2. For every $\ell \in \{2, \ldots, s+1\}$, such that $j_{\ell-1} \leq j_\ell - 2$, there exists $k \in \{1, \ldots, \ell - 1\}$ such that the segment $M_{j_k} M_{j_k+1} \cdots M_{j_\ell - 2}$ is stable.

Proof of Claim. The condition $j_{\ell-1} \leq j_\ell - 2$ implies that $j_{\ell-1} < j_\ell - 1 < j_\ell$.



By construction of the greedy basis of $\{\iota(\mathrm{im}\, M_i) \mid 1 \leq i < m\}$, for every index $i < j_\ell$, the vector $\iota(\mathrm{im}(M_i))$ is a linear combination of $\iota(\mathrm{im}(M_{j_p}))$ where $j_p \in \{j_1, \ldots, j_{\ell-1}\}$. Since $j_{\ell-1} < j_\ell - 1 < j_\ell$, it follows that there exists a nonzero vector $(c_1, \ldots, c_{\ell-1}) \in \mathbb{Q}^{\ell-1}$ where

$$\iota(\mathrm{im}(M_{j_\ell - 1})) = \sum_{p=1}^{\ell-1} c_p \, \iota(\mathrm{im}(M_{j_p})) \,. \tag{3}$$

By Claim 7.1, $\mathrm{im}(M_{j_\ell - 1}) \cap \ker(M_{j_\ell - 2}) = \{0\}$, which implies $\iota(\mathrm{im}(M_{j_\ell - 1})) \wedge \iota(\ker(M_{j_\ell - 2})) \neq 0$. Substituting the linear combination in (3) and bilinearity of the wedge product yields

$$\sum_{p=1}^{\ell-1} c_p \left(\iota(\mathrm{im}(M_{j_p})) \wedge \iota(\ker(M_{j_\ell - 2}))\right) \neq 0.$$

Let $k \in \{1, \ldots, \ell - 1\}$ be an index such that $c_k(\iota(\mathrm{im}(M_{j_k})) \wedge \iota(\ker(M_{j_\ell - 2}))) \neq 0$, giving that $\mathrm{im}(M_{j_k}) \cap \ker(M_{j_\ell - 2}) = 0$. Let $P = M_{j_k} \ldots M_{j_\ell - 2}$, and observe that $\mathrm{rk}(P) = r$ by Claim 7.1. Since $\mathrm{rk}(P) = \mathrm{rk}(M_{j_k})$ we have $\mathrm{im}(P) = \mathrm{im}(M_{j_k})$, and likewise since $\mathrm{rk}(P) = \mathrm{rk}(M_{j_\ell - 2})$ we have $\ker(P) = \ker(M_{j_\ell - 2})$. As a result, $\mathrm{im}(P) \cap \ker(P) = \{0\}$ and P is stable. ♣

Now we are in a position to state the following claim:

[1] The definition of $\iota(W)$ depends up to a nonzero scalar multiple on the chosen basis of W.

Claim 7.3. For $\ell \in \{2, \ldots, s+1\}$ there exists $k < \ell$ such that the segment $M_{j_k}, \ldots, M_{j_{\ell-1}}$ has a factorization tree of height at most 2.

Proof of Claim. The claim holds if $j_{\ell-1} = j_\ell - 1$ since then we take $k := \ell - 1$ and the segment $M_{j_k}, \ldots, M_{j_{\ell-1}}$ is just a single matrix. In the case that $j_{\ell-1} < j_\ell - 1$, by Claim 7.2, there exists $k < \ell$ such that $P := M_{j_k} \cdots M_{j_\ell - 2}$ is stable. It follows that $P \cdot M_{j_\ell - 1}$ has a factorization tree of height 2, as shown in Figure 2. ♣

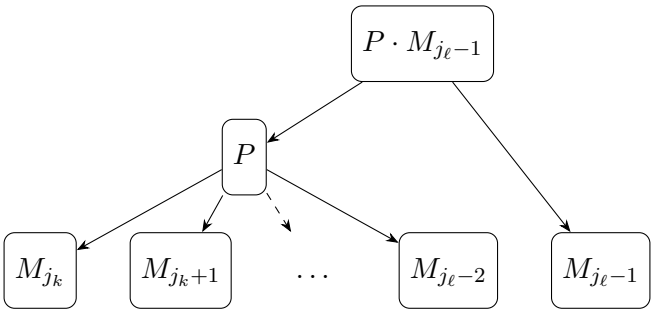


Figure 2: factorization tree of $M_{j_k}, \ldots, M_{j_\ell-1}$ with height at most 2

Since each such segment $M_{j_k}, \ldots, M_{j_{\ell-1}}$ has a factorization tree of height at most 2, and by (2) there are at most $s \leq 2^d$ such segments, we can combine their trees using a binary tree of height at most $\lceil \log_2 s \rceil \leq d$. Therefore, the final tree has height at most $d + 2$, as desired. □

Proposition 3. Every sequence of matrices in $M_d(\mathbb{Q})$ is the yield of some factorization tree of height at most $\theta(d) := d(d+3)$.

Proof. We construct the factorization tree in successive overlapping *strata*, where each *stratum* comprises at most three levels of the final tree and the top level of a given stratum is the lowest level of the next stratum above. The construction proceeds bottom-up and is such that the labels of all nodes in stratum ℓ, except possibly the rightmost node in each level, have rank at most $d - \ell$. The lowest level of the base stratum, $\ell = 0$, contains the leaf nodes corresponding to the input sequence of matrices (see Figure 3).

We now describe how to construct a given stratum given list $M_1, \ldots, M_m$ at its bottom level. We group these nodes to form products $N_1, \ldots, N_s$, where each product N_j is defined as

$$N_j = M_{i_j} M_{i_j+1} \cdots M_{i_{j+1}-1}, \qquad \text{for } j = 1, \ldots, s, \tag{4}$$

with indices $1 = i_1 < i_2 < \cdots < i_{s+1} = m + 1$ chosen so that the rank of each product N_j is equal to the rank of all its factors:

$$\mathrm{rk}(N_j) = \mathrm{rk}(M_{i_j}) = \mathrm{rk}(M_{i_j+1}) = \ldots = \mathrm{rk}(M_{i_{j+1}-1}), \quad \text{for all } j.$$

We further require that the indices $i_1, \ldots, i_s$ are lexicographically maximal with the above property. This means that the product N_j includes as many consecutive M_k's as possible, starting from M_{i_j}, subject to rank of the product being equal to $\mathrm{rk}(M_{i_j})$. It follows that for each $j \in \{1, \ldots, s-1\}$, the product $N_j N_{j+1}$ has strictly smaller rank than either N_j or N_{j+1}. That is,

$$\mathrm{rk}(N_j N_{j+1}) < \max(\mathrm{rk}(N_j), \mathrm{rk}(N_{j+1})),$$

otherwise, we would have $\mathrm{rk}(N_j) = \mathrm{rk}(N_{j+1}) = \mathrm{rk}(N_j N_{j+1})$ which contradicts the lexicographic maximality of $i_1, \ldots, i_s$.

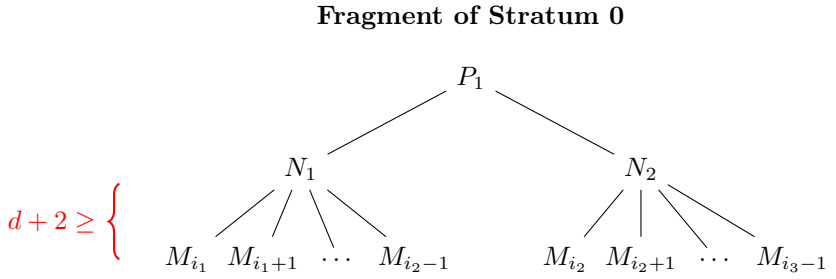


Figure 3: Fragment of Stratum 0, showing two products N_1 and N_2 paired into a parent node P_1.

To build the third level, we combine the N_j's in adjacent pairs: if s is even, we form the products N_1N_2, N_3N_4, ..., $N_{s-1}N_s$; and if s is odd, we form the products N_1N_2, N_3N_4, ..., $N_{s-2}N_{s-1}$, N_s. Observe that, for all j,

$$\mathrm{rk}(N_jN_{j+1}) < \max(\mathrm{rk}(N_j), \mathrm{rk}(N_{j+1})) \leq d - \ell\,.$$

Since the top-most level in a stratum is the bottom-most level in next stratum, for all nodes P in the next stratum, except possibly the rightmost one, we have $\mathrm{rk}(P) \leq d - (\ell + 1)$. Therefore, the rank bound is preserved in the new stratum, except possibly at the rightmost node (when s is odd).

By Proposition 7, each product N_j, as in equation (4), can be computed using a factorization tree of height at most $d+2$, with leaves labeled $M_{i_j}, \ldots, M_{i_{j+1}-1}$. Since there are at most d strata, we obtain an overall bound of $d(d+3)$ for the total height of the factorization tree. □

We furthermore prove the following result, which is crucial in our study of Cover Closure and Reach Closure. Let $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$ be monoid morphisms, where $\mathbb{Z}$ is considered additively and $\omega(\sigma) \in \{-1, 0, 1\}$ for all $\sigma \in \Sigma$.

Proposition 4. There exists a threshold $\eta(d) := 2^{d(d+3)} + 1$ such that for all $w \in \Sigma^*$:

(i) If $\omega(w) = \eta(d)$, then there exists a factor u of w such that $\omega(u) > 0$ and $\varphi(u)$ is stable.

(ii) If $\omega(w) = -\eta(d)$, then there exists a factor u of w such that $\omega(u) < 0$ and $\varphi(u)$ is stable.

Proof. By symmetry, it suffices to prove the first item. We prove the contrapositive, that is, we assume for every factor u of w such that $\varphi(u)$ is stable we have $\omega(u) \leq 0$, and argue that $\omega(w) < \eta(d)$. To this end, write $w = w_1w_2\cdots w_m$ and consider a factorization tree of the sequence of matrices $\varphi(w_1), \ldots, \varphi(w_m)$. By Proposition 3 there is such a tree $\mathcal{T}$ of height at most $d(d+3)$.

For each node α of $\mathcal{T}$ we define corresponding string $u_\alpha \in \Sigma^*$ bottom-up from the leaves. For α the i-th leaf node we have $u_\alpha := w_i$. For a non-leaf node α with children $\alpha_1, \ldots, \alpha_\ell$, we have $u_\alpha = u_{\alpha_1} \cdots u_{\alpha_\ell}$. Clearly we have $u_\alpha = w$ for the root node α.

We claim that for a node α of height h we have $\omega(u_\alpha) \leq 2^h$. The proof is by induction on h. The base case is that α is a leaf. Then $u_\alpha = \sigma$ for some $\sigma \in \Sigma$, so $\omega(u_\alpha) \in \{-1, 0, 1\}$, and hence $\omega(u_\alpha) \leq 2^0 = 1$. For the inductive step, suppose α is a binary node with children α_1 and α_2 at height h. Then

$$\omega(u_\alpha) = \omega(u_{\alpha_1}) + \omega(u_{\alpha_2}) \leq 2 \cdot 2^{h-1} = 2^h\,.$$

On the other hand, if α has three or more children then $\varphi(u_\alpha)$ (the label of u_α) is stable by definition of a factorization tree and hence $\omega(u_\alpha) \leq 0$ by assumption. Thus the induction hypothesis also holds in this case, and the claim is established.

Since the root node α has height at most $d(d+3)$ and satisfies $u_\alpha = w$, it follows from the above claim that

$$\omega(w) = \omega(u_\alpha) \leq 2^{d(d+3)} < \eta(d).$$

□

4 Algebraic Geometry Background

We recall basic notions from algebraic geometry, and refer the reader to [22, 4] for an extended background.

An *affine variety* (or *algebraic set*) $X \subseteq \overline{\mathbb{Q}}^d$ is the set of common zeros of finitely many polynomials $P_1, \ldots, P_\ell \in \overline{\mathbb{Q}}[x_1, \ldots, x_d]$, that is,

$$X = \left\{ x \in \overline{\mathbb{Q}}^d : P_1(x) = \cdots = P_\ell(x) = 0 \right\}.$$

Given an ideal $I \subseteq \overline{\mathbb{Q}}[x_1, \ldots, x_d]$, the associated variety is $V(I) = \{x \in \overline{\mathbb{Q}}^d : \forall p \in I, P(x) = 0\}$. By Hilbert's Basis Theorem, every ideal I is finitely generated. Affine varieties are precisely the sets of the form $V(I)$ for some ideal $I \subseteq \overline{\mathbb{Q}}[x_1, \ldots, x_d]$.

The *Zariski topology* on $\overline{\mathbb{Q}}^d$ takes varieties as closed sets. For $S \subseteq \overline{\mathbb{Q}}^d$, we denote by $\overline{S}$ its Zariski closure in the topology, that is, the smallest algebraic set containing S. Polynomial maps are continuous in Zariski topology, meaning that $f(\overline{S}) \subseteq \overline{f(S)}$ holds for all sets $S \subseteq \overline{\mathbb{Q}}^d$ and polynomial maps $f : \overline{\mathbb{Q}}^d \to \overline{\mathbb{Q}}^{d'}$. The following fact is immediate, see a proof in Section E.

Fact 8. Let $f : X \to Y$ be a continuous function of topological spaces X and Y. Then $\overline{f(S)} = \overline{f(\overline{S})}$ for every subset $S \subseteq X$.

The Zariski topology is *Noetherian*: every descending chain of closed sets stabilizes. A set is irreducible if it can not be written as the union of two proper closed subsets. every closed set decomposes uniquely as a finite union of irreducible components—that is, maximal (respective to set inclusion) irreducible closed subsets. The *dimension* of a variety X is the maximal length of a strict chain of irreducible closed subsets in X; when $X \subseteq \overline{\mathbb{Q}}^d$ we have $\dim(X) \leq d$.

In this paper, the two primary varieties of interest are $M_d(\overline{\mathbb{Q}}) \simeq \overline{\mathbb{Q}}^{d^2}$ and

$$\mathrm{GL}_d(\overline{\mathbb{Q}}) = \{(A, y) \in \overline{\mathbb{Q}}^{d^2+1} : \det(A) \cdot y = 1\},$$

viewed as an affine variety via the polynomial relation $\det(A) \cdot y = 1$. A *linear algebraic group* $G \subseteq \mathrm{GL}_d(\overline{\mathbb{Q}})$ is a group that is also an affine variety. The group operations (multiplication and inversion) are polynomial maps— and hence are continuous with respect to the Zariski topology.

The following is a classical result, see for example [5, 1.2. Consequence]; for completeness, we include a proof in Section E.

Fact 9. A closed subsemigroup of $GL_d(\overline{\mathbb{Q}})$ is a linear algebraic group.

Stable Matrices. The following lemma on the Zariski closure of monoids generated by stable matrices is a variant of notions previously explored in [19, Proposition 11], adapted to our setting. See Section E for a detailed proof.

Lemma 10. Let $M \in M_d(\mathbb{Q})$ be stable. Then $\overline{\{M^n : n > 0\} \cap \{N : \mathrm{rk}(N) = \mathrm{rk}(M)\}}$ is a group.

5 Algorithm for Cover Closure

In this section, we reduce Cover Closure to the problem of computing the Zariski closure of a finitely generated matrix monoid, for which there is a known algorithm [20].

Proposition 11. There is a reduction from Cover Closure to the problem of computing the Zariski closure of a finitely generated matrix monoid.

Proof. Consider an instance of Cover Closure, given by monoid morphisms $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$ with $\omega(\sigma) \in \{-1, 0, 1\}$ for all $\sigma \in \Sigma$. The problem asks to compute the Zariski closure of $\varphi(L_C(\omega))$, where $L_C(\omega) = \{w \in \Sigma^* : \forall u \preceq w \cdot \omega(u) \geq 0\}$.

We organize the proof around two key technical claims. At a high level, the first claim constructs an NFA $\mathcal{A}$ whose language $L(\mathcal{A})$ is a superset of $L_C(\omega)$. The second claim shows that this over-approximation is sufficiently tight such that $\varphi(L_C(\omega))$ and $\varphi(L(\mathcal{A}))$ have the same Zariski closure.

Claim 11.1. We can compute an NFA $\mathcal{A}$ such that

1. $L_C(\omega) \subseteq L(\mathcal{A})$;
2. every word $w \in L(\mathcal{A}) \setminus L_C(\omega)$ admits a factorization $w = w_1 u w_2$, such that $\varphi(u)$ is stable, $\omega(u) > 0$, and $\omega(w') \geq 0$ for every prefix w' of $w_1 u$.

Proof of Claim. The NFA $\mathcal{A}$ has set of states $Q := \{0, 1, \ldots, \eta(d) - 1\} \cup \{\infty\}$ for $\eta(d) = 2^{d(d+3)} + 1$, as defined in Proposition 4. State 0 is initial and all states are accepting. Intuitively, the states of $\mathcal{A}$ keep track the accumulated weight of the input word up to $\eta(d) - 1$, thereafter treating the weight as infinite. The automaton blocks any symbol σ with negative weight while in state 0. Reading a symbol with positive weight in state $\eta(d) - 1$ leads to state ∞, which is a sink. Formally, the transition relation $\Delta \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}$ is given by

- $(c, \sigma, c + \omega(\sigma)) \in \Delta$ if $c, c + \omega(\sigma) \in \{0, \ldots, \eta(d) - 1\}$;
- $(\eta(d) - 1, \sigma, \infty) \in \Delta$ if $\omega(\sigma) = 1$;
- $(\infty, \sigma, \infty) \in \Delta$ for all $\sigma \in \Sigma$.

It is straightforward that $L_C(\omega) \subseteq L(\mathcal{A})$, which is Item 1 of the claim. For Item 2, let $w \in L(\mathcal{A}) \setminus L_C(\omega)$. Then w is accepted along a run of $\mathcal{A}$ that enters the state ∞ at some point (otherwise the state of $\mathcal{A}$ exactly records the accumulated weight along the word w, ensuring that $w \in L_C(\omega)$). For such a run, let v be the prefix of w leading to the first visit of state ∞. Then $\omega(v) = \eta(d)$. By Proposition 4 the word v admits a factorization $v = w_1 u w_2$ with $\varphi(u)$ stable and $\omega(u) > 0$. We furthermore have $\omega(w') \geq 0$ for every prefix w' of v by construction of $\mathcal{A}$. This yields the desired factorization of Item 2. ♣

We next use Claim 11.1 to show that $\varphi(L(\mathcal{A}))$ and $\varphi(L_C(\omega))$ have the same Zariski closure.

Claim 11.2. If $\mathcal{A}$ satisfies Conditions 1 and 2 of Claim 11.1 then $\overline{\varphi(L_C(\omega))} = \overline{\varphi(L(\mathcal{A}))}$.

Proof of Claim. Since $L_C(\omega) \subseteq L(\mathcal{A})$, it suffices to show that for every word $w \in L(\mathcal{A}) \setminus L_C(\omega)$ we have

$$\varphi(w) \in \overline{\varphi(L_C(\omega))}\,.$$

To this end, consider the factorization $w = w_1 u w_2$ arising from Item 2 of Claim 11.1. Since $\omega(u) > 0$ and $\omega(w') \geq 0$ for all prefixes w' of $w_1 u$, there exists n_0 such that for all $n \geq n_0$ we have

$w_1u^nw_2 \in L_C(\omega)$—specifically for n sufficiently large it holds that $\omega(w') \geq 0$ for all prefixes w' of $w_1u^nw_2$.

Since $\varphi(u)$ is stable, by Lemma 10 $\overline{\langle\varphi(u)\rangle}$ is a group. By Fact 9, the subsemigroup $\overline{\{\varphi(u^n) : n \geq n_0\}}$ is the group $\overline{\langle\varphi(u)\rangle}$. Thereby contains a matrix I_u such that $I_u\varphi(u) = \varphi(u)I_u = \varphi(u)$. The following multiplication map is Zariski continuous

$$\begin{aligned} f : M_d(\overline{\mathbb{Q}}) &\to M_d(\overline{\mathbb{Q}}) \\ A &\mapsto \varphi(w_1u)\, A\, \varphi(w_2), \end{aligned}$$

Since $f(I_u) = \varphi(w_1uw_2) = \varphi(w)$, we get

$$\begin{aligned} \varphi(w) &\in f\left(\overline{\{\varphi(u)^n : n \geq n_0\}}\right) \\ &\subseteq \overline{f\left(\{\varphi(u)^n : n \geq n_0\}\right)} = \overline{\{\varphi(w_1u^{1+n}w_2) : n \geq n_0\}} \\ &\subseteq \overline{\varphi(L_C(\omega))}, \end{aligned} \tag{5}$$

where the middle inclusion follows by continuity of f. ♣

This completes the reduction and its correctness proof. □

By [20, Theorem 3] there is a procedure to compute $\overline{\varphi(L(\mathcal{A}))}$ given an NFA $\mathcal{A}$ and morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$. In combination with Proposition 11, this gives the decidability of the Cover Closure problem.

6 Algorithm for Reach Closure

In this section we give a procedure to decide Reach Closure. There are two steps. In the first step, we reduce Reach Closure to the problem Zero Closure, which we introduce below, and in the second step we give a procedure for the problem Zero Closure.

6.1 Reduction from Reach Closure to Zero Closure

For a morphism $\omega : \Sigma^* \to \mathbb{Z}$, define

$$L_Z(\omega) := \{w \in \Sigma^* : \omega(w) = 0\}$$

to be the set of words of weight zero. The Zero Closure problem asks, given monoid morphisms $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$ with $\omega(\sigma) \in \{-1, 0, 1\}$ for all $\sigma \in \Sigma$, compute $\overline{\varphi(L_Z(\omega))}$.

Proposition 12. Reach Closure reduces to Zero Closure.

Proof. Consider morphisms $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$. Recall that the goal of Reach Closure is to compute the closure of $\varphi(L_R(\omega))$. The following is the key technical claim.

Claim 12.1. We can compute a finite automaton $\mathcal{A}$ such that, writing $L' := L(\mathcal{A}) \cap L_Z(\omega)$, we have $L_R(\omega) \subseteq L'$ and every word $w \in L' \setminus L_R(\omega)$ admits a factorization $w = w_1uw_2vw_3$, where

1. $\varphi(u)$ is stable, the weight of each prefix of w_1u is nonnegative and $\omega(u) > 0$;
2. $\varphi(v)$ is stable, the weight of each suffix of vw_3 is nonpositive and $\omega(v) < 0$.

Proof of Claim. Automaton $\mathcal{A}$ has set of states

$$Q := (\{0, 1, \ldots, \eta(d) - 1\} \times \{0, 1\}) \cup \{\infty\}, \tag{6}$$

where $\eta(d) = 2^{d(d+3)} + 1$ is as defined in Proposition 4. Intuitively, the states of $\mathcal{A}$ keep track of the total weight of the word read so far, up to weight $\eta(d) - 1$. While words in the language $L' := L(\mathcal{A}) \cap L_Z(\omega)$ can have negative-weight prefixes, the construction of $\mathcal{A}$ ensures that any negative-weight prefix of a word $w \in L'$ is sandwiched between the respective shortest and longest prefixes of w of weight $\eta(d)$. This is the key to proving the decomposition featured in the claim.

Formally, automaton $\mathcal{A}$ has alphabet Σ, set of states Q as shown in (6), initial state $(0, 0)$, and accepting states $\{(0, 0), (0, 1)\}$. The set $\Delta \subseteq Q \times \Sigma \times Q$ of transitions is defined such that

- for all $\sigma \in \Sigma$ and $b \in \{0, 1\}$, there is a transition $((c, b), \sigma, (c + \omega(\sigma), b))$ provided that $c, c + \omega(\sigma) \in \{0, \ldots, \eta(d) - 1\}$;
- for all $\sigma \in \Sigma$ there is a transition (∞, σ, ∞);
- for all $\sigma \in \Sigma$ with $\omega(\sigma) = 1$ there is a transition $((\eta(d) - 1, 0), \sigma, \infty)$;
- for all $\sigma \in \Sigma$ with $\omega(\sigma) = -1$ there is transition $(\infty, \sigma, (\eta(d) - 1, 1))$.

We first argue that $L_R(\omega) \subseteq L'$, that is, $\mathcal{A}$ accepts any word of zero weight all of whose prefixes have non-negative weight. Given a word $w \in L_R(\omega)$, suppose first that there is no prefix with weight $\eta(d)$. Then there is a run of $\mathcal{A}$ on w that never enters the state ∞ and exactly records the accumulated weight along the word w. Otherwise, let x and y be, respectively, the shortest and longest prefixes of w with weight $\eta(d)$. Then there is an accepting run of $\mathcal{A}$ on w in which the state is $(c, 0)$ after reading any proper prefix of x of weight $c < \eta(d)$; the state is ∞ after reading any prefix of w containing x and contained in y; the state is $(c, 1)$ after reading any prefix of w of weight c containing y.

We next prove Items 1 and 2 of the claim. Consider a word $w \in L' \setminus L_R(\omega)$ and an accepting run of $\mathcal{A}$ on w. Since $w \notin L_R(\omega)$ this run ends in $(0, 1)$. Hence, w has a shortest prefix x with weight $\eta(d)$. Applying Item 1 of Proposition 4, we obtain a decomposition $x = x_1 u x_2$ where $\varphi(u)$ is stable and $\omega(u) > 0$. We moreover have that $\omega(w') \geq 0$ for any prefix w' of x since the construction of $\mathcal{A}$ prevents the weight of such prefix being negative. Let y be the shortest suffix of w with weight $-\eta(d)$. Applying Item 2 of Proposition 4, we have a decomposition $y = y_1 v y_2$ where $\varphi(v)$ is stable and $\omega(v) < 0$. In addition, $\omega(w') \leq 0$ for any suffix w' of y by construction of $\mathcal{A}$. This concludes the proof of the claim. ♣

The following shows how Claim 12.1 can be used to compute the Zariski closure.

Claim 12.2. If $\mathcal{A}$ satisfies Conditions 1–2 of Claim 12.1 then $\overline{\varphi(L_R(\omega))} = \overline{\varphi(L')}$.

Proof of Claim. Since $L_R(\omega) \subseteq L'$, it suffices to show that for every word $w \in L' \setminus L_R(\omega)$ we have $\varphi(w) \in \overline{\varphi(L_R(\omega))}$. For each such word w, consider the factorization $w = w_1 u w_2 v w_3$ arising from Claim 12.1.

Since $\omega(w) = 0$, $\omega(u) > 0$, and $\omega(v) < 0$, there are infinitely many $m, n \in \mathbb{N}$ such that

$$\omega(w_1 u^m w_2 v^n w_3) = 0.$$

Moreover since every prefix of $w_1 u$ has nonnegative weight and every suffix of $v w_3$ has nonpositive weight, for m, n sufficiently large we have that $w_1 u^m w_2 v^n w_3 \in L_R(\omega)$. We may thus define an infinite semigroup

$$S := \{(m, n) \in \mathbb{N}^2 : w_1 u^{m+1} w_2 v^{n+1} w_3 \in L_R(\omega)\}.$$

Since $\varphi(u)$ and $\varphi(v)$ are stable, the matrix

$$M := \begin{pmatrix} \varphi(u)^m & 0 \\ 0 & \varphi(v)^n \end{pmatrix}$$

is also stable. By Lemma 10, the Zariski-closed sub-semigroup $\overline{\langle M \rangle}$ is a group. Hence,

$$\overline{\{(\varphi(u)^m, \varphi(v)^n) : (m,n) \in S\}} \subseteq M_d(\overline{\mathbb{Q}}) \times M_d(\overline{\mathbb{Q}})$$

is also a group, and contains a matrix (I_u, I_v) where I_u is such that $I_u\varphi(u) = \varphi(u)I_u = \varphi(u)$ and I_v is such that $I_v\varphi(v) = \varphi(v)I_v = \varphi(v)$. The following multiplication map is Zariski continuous

$$\begin{aligned} f : M_d(\overline{\mathbb{Q}}) \times M_d(\overline{\mathbb{Q}}) &\to M_d(\overline{\mathbb{Q}}) \\ (A, B) &\mapsto \varphi(w_1\, u)\, A\, \varphi(w_2)\, B\, \varphi(v\, w_3), \end{aligned}$$

Since $f(I_u, I_v) = \varphi(w_1\, u\, w_2\, v\, w_3) = \varphi(w)$, by a similar argument to (5), we get

$$\varphi(w) = \varphi(w_1 u w_2 v w_3) \in \overline{\{\varphi(w_1 u^{m+1} w_2 v^{n+1} w_3) : (m,n) \in S\}} \subseteq \overline{\varphi(L_R(\omega))}\,.$$

♣

Thus, the reduction and its correctness proof follows from Section B. □

6.2 Computing the Zero-Weight Language

We show in this section how to compute the Zariski closure $\overline{\varphi(L_Z(\omega))}$. The key idea is to represent $\overline{\varphi(L_Z(\omega))}$ as the image of the Zariski closure of a regular set over a product alphabet. The proof uses the following bounded variant of $L_Z(\omega)$:

$$L_{BZ}(\omega) := \{w \in \Sigma^* : \omega(w) = 0 \wedge \forall u \preceq w\, (-\eta(d) \leq \omega(u) \leq \eta(d))\}\,.$$

This is the language of words of weight 0 such that the weight of every prefix lies in $\{-\eta(d), \ldots, \eta(d)\}$.

Proposition 13. The Zero Closure problem is decidable, that is, there is a procedure that given morphisms $\omega : \Sigma^* \to \mathbb{Z}$ and $\varphi : \Sigma^* \to M_d(\mathbb{Q})$, computes $\overline{\varphi(L_Z(\omega))}$.

Proof. Consider the alphabet $\Gamma = (\Sigma \cup \{\varepsilon\})^4 \setminus \{\varepsilon\}^4$. Define the map $\mathrm{prod} : \Gamma^* \to (\Sigma^*)^4$ by

$$\mathrm{prod}((a_1,b_1,c_1,d_1)\cdots(a_m,b_m,c_m,d_m)) := (a_1\cdots a_m, b_1\cdots b_m, c_1\cdots c_m, d_1\cdots d_m)$$

and define $\mathrm{flat} : \Gamma^* \to \Sigma^*$ by

$$\mathrm{flat}((a_1,b_1,c_1,d_1)\cdots(a_m,b_m,c_m,d_m)) := a_1\cdots a_m\, b_1\cdots b_m\, c_1\cdots c_m\, d_1\cdots d_m\,.$$

For a word $W \in \Gamma^*$ such that all but one component of $\mathrm{prod}(W)$ is ε, we may safely identify W with $\mathrm{prod}(W)$, since the latter uniquely determines the former.

We construct a finite-state automaton $\mathcal{A}$ over the alphabet Γ such that

$$\overline{\varphi(L_Z(\omega))} = \overline{\varphi(L_{BZ}(\omega) \cup \mathrm{flat}(L(\mathcal{A})))}\,. \tag{7}$$

Since $L_{BZ}(\omega)$ is a regular language, the closure of its image under φ is computable.

Claim 13.1. $\overline{\varphi(\mathrm{flat}(L(\mathcal{A})))}$ is computable.

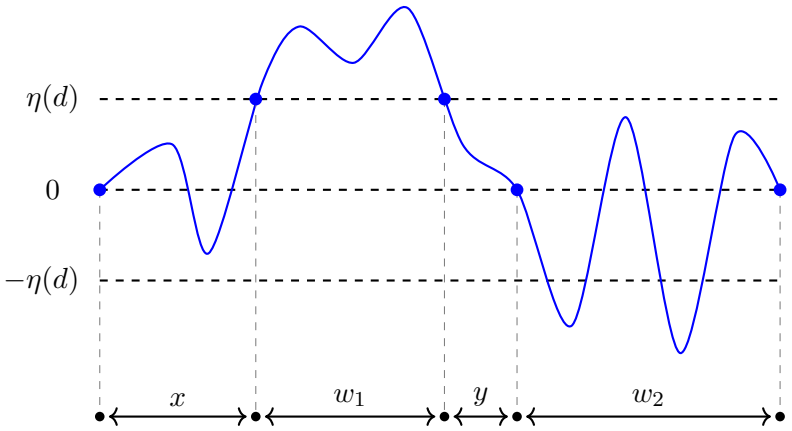


Figure 4: Decomposition $w = xw_1yw_2$, where x is the shortest prefix of w with weight in $\{-\eta(d), \eta(d)\}$, the infix xw_1y is the shortest weight-zero prefix of w that extends x, and y is the shortest suffix of xw_1y with weight $\omega(y) = -\omega(x)$.

Proof of Claim. Consider the map ψ that sends a word $w = (w_1, w_2, w_3, w_4) \in L(\mathcal{A})$ to the block diagonal matrix $\mathrm{diag}(\varphi(w_1), \varphi(w_2), \varphi(w_3), \varphi(w_4))$. By regularity of $L(\mathcal{A})$, it follows that $\overline{\psi(L(\mathcal{A}))}$ is computable. The computability of $\overline{\varphi(\mathrm{flat}(L(\mathcal{A})))}$ follows from a simple observation—namely, the fact that, $\overline{\varphi(\mathrm{flat}(L(\mathcal{A})))}$ is the closure of the image of $\overline{\psi(L(\mathcal{A}))}$ under the polynomial map that multiplies the diagonal blocks. ♣

The set of states of $\mathcal{A}$ is $\{-2\eta(d), \ldots, 2\eta(d)\}$. The state 0 is initial and is the unique accepting state. There is a transition from $q \in Q$ to $q' \in Q$ with label $(a, b, c, d) \in \Gamma$ precisely when $q' = q + \omega(abcd)$. By construction, for all words $W \in L(\mathcal{A})$, after reading W the state of $\mathcal{A}$ is $\omega(\mathrm{flat}(W)) = 0$. Thus $\mathrm{flat}(L(\mathcal{A})) \subseteq L_Z(\omega)$, which establishes the right-to-left inclusion in (7).

It remains to prove the left-to-right inclusion in (7). Suppose that $w \in L_Z(\omega)$. We show that if $w \notin L_{BZ}(\omega)$ then $\varphi(w) \in \overline{\varphi(\mathrm{flat}(L(\mathcal{A})))}$. To this end, we will construct two words $W, U \in \Gamma^*$ such that $WU^k \in L(\mathcal{A})$ for all $k \in \mathbb{N}$, and moreover

$$\varphi(w) \in \overline{\varphi(\{\mathrm{flat}(WU^k) \,:\, k \in \mathbb{N}\})}\,. \tag{8}$$

By the assumption that $w \notin L_{BZ}(\omega)$ there is a prefix $x \preceq w$ such that $\omega(x) \in \{-\eta(d), \eta(d)\}$. Let x be the shortest such prefix. We suppose that $\omega(x) = \eta(d)$; the argument for the case $\omega(x) = -\eta(d)$ is symmetric. Then w admits a decomposition $w = xw_1yw_2$, shown in Figure 4, where

- xw_1y is the shortest prefix of w containing x with weight 0,
- y is the shortest suffix of xw_1y with weight $-\eta(d)$, and
- $\omega(w_1) = \omega(w_2) = 0$.

Applying Proposition 4 we have decompositions $x = x_1ux_2$ and $y = y_2vy_1$ such that $\varphi(u)$ and $\varphi(v)$ are stable, $\omega(u) > 0$, and $\omega(v) < 0$.

In summary, we have

$$w = \underbrace{x_1\,u\,x_2}_{\eta(d)}\ \underbrace{w_1}_{0}\ \underbrace{y_2\,v\,y_1}_{-\eta(d)}\ \underbrace{w_2}_{0}$$

with $\varphi(u), \varphi(v)$ stable, $\omega(u) > 0$, and $\omega(v) < 0$. From this decomposition of w we construct a word $W \in L(\mathcal{A})$ such that

$$\text{flat}(W) = x_1\, u^{1+m}\, x_2\, w_1\, y_2\, v^{1+n}\, y_1\, w_2\,, \tag{9}$$

where $m, n \in \mathbb{N}$. We construct $W = W_1W_2W_3$ as the concatenation of three words $W_1, W_2, W_3 \in \Gamma^*$ such that

1. $\text{prod}(W_1) = (x_1u, x_2, y_2v, y_1)$;
2. $\text{prod}(W_2) = (u^{m_0}, w_1, v^{n_0}, w_2)$;
3. $\text{prod}(W_3) = (u^{m_1}, \varepsilon, v^{n_1}, \varepsilon)$

for $m_0, m_1, n_0, n_1 \in \mathbb{N}$. Below we specify the words W_1, W_2, W_3 and simultaneously describe an accepting run of $\mathcal{A}$ over W.

We divide the run into three phases, according to the factor that is being read.

- **Phase 1.** In this phase we read W_1. To unambiguously specify the word W_1 we stipulate that $W_1 = (x_1u, \varepsilon, \varepsilon, \varepsilon) \cdot (\varepsilon, x_2, \varepsilon, \varepsilon) \cdot (\varepsilon, \varepsilon, y_2v, \varepsilon) \cdot (\varepsilon, \varepsilon, \varepsilon, y_1)$. Since $x_1ux_2y_2vy_1 \in L_{BZ}(\omega)$, the automaton $\mathcal{A}$ can read W_1, starting and ending in state 0 with all intermediate states lying in $\{-\eta(d), \ldots, \eta(d)\}$; see Figure 4.

- **Phase 2.** In this phase we read W_2. At the start of Phase 2 the automaton is in state 0. Suppose that $w_1 = \sigma_1 \cdots \sigma_\ell$ and $w_2 = \xi_1 \cdots \xi_k$. The aim is to read the sequence of symbols $(\varepsilon, \sigma_i, \varepsilon, \varepsilon)$ for $i = 1, \ldots, \ell$ followed by the sequence of symbols $(\varepsilon, \varepsilon, \varepsilon, \xi_i)$ for $i = 1, \ldots, k$. At the same time, we aim to keep the automaton state in the bounded range $\{-2\eta(d), \ldots, 2\eta(d)\}$. This can be done by interleaving the input with factors $(u, \varepsilon, \varepsilon, \varepsilon)$ and $(\varepsilon, \varepsilon, v, \varepsilon)$ according to the following rule: If ever the state equals $-\omega(u)$ then read $(u, \varepsilon, \varepsilon, \varepsilon)$ to bring the state back to 0. Likewise, if ever the state equals $-\omega(v)$, then read $(\varepsilon, \varepsilon, v, \varepsilon)$ to bring the state back to 0. See Figure 5 for an illustration of such an interleaving. The numbers m_0, n_0 in the description of W_2 record the respective number of times we consume the factors $(u, \varepsilon, \varepsilon, \varepsilon)$ and $(\varepsilon, \varepsilon, v, \varepsilon)$ in Phase 2.

- **Phase 3.** In this phase we read W_3. At the start of Phase 3 the state of the automaton $\mathcal{A}$ is in the range $\{-\eta(d), \ldots, \eta(d)\}$. Moreover, since $\omega(w) = 0$, we have

  $$\omega(\text{flat}(W_1W_2)) = m_0\omega(u) + n_0\omega(v)\,.$$

  Now the equation $m\omega(u) + n\omega(v) = 0$ has infinitely many solutions in positive integers m, n. Pick one such solution (m, n) such that $m_0 \le m$ and $n_0 \le n$.
  Writing $m_1 := m - m_0$ and $n_1 := n - n_0$, our aim in Phase 3 is to read W_3 such that

  $$\text{prod}(W_3) = (u^{m_1}, \varepsilon, v^{n_1}, \varepsilon)\,.$$

  This choice of W_3 ensures that $\omega(\text{flat}(W_1W_2W_3)) = 0$. We can read W_3 while staying in the range $\{-2\eta(d), \ldots, 2\eta(d)\}$ using the following greedy procedure. Suppose that so far we have read $W_3' \in \Gamma^*$ such that $\text{prod}(W_3') = (u^{m'}, \varepsilon, v^{n'}, \varepsilon)$ with $m' \le m_1$ and $n' \le n_1$. If $m' < m_1$ and $n' < n_1$ then we read $(u, \varepsilon, \varepsilon, \varepsilon)$ if the current state is negative and we read $(\varepsilon, \varepsilon, v, \varepsilon)$ if the current state is positive. If $m' < m_1$ and $n' = n_1$ then we read $(u, \varepsilon, \varepsilon, \varepsilon)$. If $m' = m_1$ and $n' < n_1$ then we read $(\varepsilon, \varepsilon, v, \varepsilon)$.

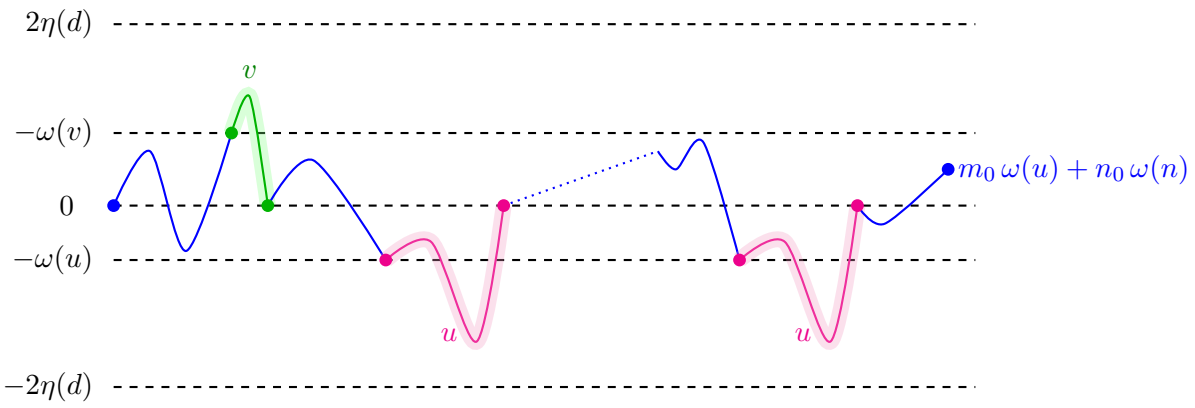


Figure 5: Depiction of the word W_2, where the vertical axis represents the prefix weight. The parts in blue represent symbols $(\varepsilon, \sigma_i, \varepsilon, \varepsilon)$ or $(\varepsilon, \varepsilon, \varepsilon, \xi_i)$. These are interleaved with words $(u, \varepsilon, \varepsilon, \varepsilon)$ and $(\varepsilon, \varepsilon, v, \varepsilon)$ so as to keep the weight in the range $\{-2\eta(d), \ldots, 2\eta(d)\}$.

This completes the construction of a word $W \in L(\mathcal{A})$ such that there exist $m, n \in \mathbb{N}$ satisfying (9). Next we claim that there exists $U \in L(\mathcal{A})$ with

$$\mathrm{prod}(U) = (u^m, \varepsilon, v^n, \varepsilon)\,.$$

We construct U and an accepting run of $\mathcal{A}$ over U in a greedy fashion exactly as described in Phase 3 above. This yields a run that stays in the range $\{-2\eta(d), \ldots, 2\eta(d)\}$. Since the unique accepting state of $\mathcal{A}$ is also the initial state, the language $L(\mathcal{A})$ is a semigroup, implying that $U^k \in L(\mathcal{A})$ for all $k \in \mathbb{N}$.

In summary we have constructed words $W, U \in \Gamma^*$ such that $WU^k \in L(\mathcal{A})$ for all $k \in \mathbb{N}$. To conclude the proof, it remains to prove (8). Observe that

$$\mathrm{flat}(WU^k) = x_1\, u^{1+(k+1)m}\, x_2\, w_1\, y_2\, v^{1+(k+1)n}\, y_1\, w_2\,,$$

holds for all $k \in \mathbb{N}$, which in turn implies that

$$\varphi(x_1\, u^{1+(k+1)m}\, x_2\, w_1\, y_2\, v^{1+(k+1)n}\, y_1\, w_2) \in \varphi(\mathrm{flat}(L(\mathcal{A}))).$$

Now $S = \{(\varphi(u)^{(k+1)m}, \varphi(v)^{(k+1)n}) : k \in \mathbb{N}\}$ is a sub-semigroup of $M_d(\mathbb{Q}) \times M_d(\mathbb{Q})$. Since $\varphi(u)$ and $\varphi(v)$ are stable, by Lemma 10 $\overline{S} \subseteq M_d(\overline{\mathbb{Q}}) \times M_d(\overline{\mathbb{Q}})$ is a group with respect to the operation of matrix multiplication on each component. In particular, $\overline{S}$ contains an element (I_u, I_v) such that $I_u\, \varphi(u) = \varphi(u)\, I_u = \varphi(u)$ and $I_v\, \varphi(v) = \varphi(v)\, I_v = \varphi(v)$. By Zariski continuity of matrix multiplication, and by a similar argument to (5), it follows that

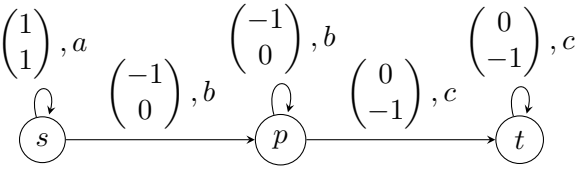
$$\varphi(w) = \varphi(x_1 u)\, I_u\, \varphi(x_2 w_1 y_2 v)\, I_v\, \varphi(y_1) \in \overline{\varphi(\mathrm{flat}(L(\mathcal{A})))}\,.$$

This completes the proof of the left-to-right inclusion in (7). □

7 Indexed Grammars

An indexed grammar G is a tuple (V, Σ, I, P, S) where V is the set of variables, Σ the set of terminals (or the alphabet), I the set of indices, $S \in V$ the starting symbol, and P is a finite set of production rules of the forms $A \to \alpha$, $A \to B\, f$ and $A\, f \to \alpha$ where $A, B \in V$, $f \in I$ and α is a word over $V \cup \Sigma$.

Derivations in an indexed grammar are similar to those in a context-free grammar except that the nonterminals can be followed by a string of indices. When a production rule $A \to BC$ is applied, the string of indices for A is attached to both B and C. See [18, Page 389] for more details; in the following we reiterate [18, Example 14.2] for the most familiar index language $\{a^n b^n c^n \mid n \geq 1\}$. We highlight that this language can be recognized as the reachability language of the following 2-VASS:



Example 6. Let $G = (V, \Sigma, I, P, S)$ be such that $V = \{S, T, A, B, C\}$, $\Sigma = \{a, b, c\}$ and $I = \{f, g\}$ and with the following set P of production rules

$$\begin{array}{lll} S \to Tg & A\,f \to a\,A & A\,g \to a \\ T \to T\,f & B\,f \to b\,B & B\,g \to b \\ T \to A\,B\,C & C\,f \to c\,C & C\,g \to c \end{array}$$

An example derivation is

$$\begin{aligned} S \Rightarrow T\,g \Rightarrow T\,f\,g \Rightarrow A\,f\,g\,B\,f\,g\,C\,f\,g \Rightarrow a\,A\,g\,B\,f\,g\,C\,f\,g \Rightarrow a^2\,B\,f\,g\,C\,f\,g \\ \Rightarrow a^2\,b\,B\,g\,C\,f\,g \Rightarrow a^2\,b^2\,C\,f\,g \Rightarrow a^2\,b^2\,c\,C\,g \Rightarrow a^2\,b^2\,c^2 \end{aligned}$$

In the above derivation, if the production rule $T \to T\,f$ at the second step repeats $n-1$ times the word $a^n\,b^n\,c^n$ is derived. ◀

Given an indexed grammar G over a finite alphabet Σ and $\varphi : \Sigma^* \to \mathrm{M}_d(\mathbb{Q})$ a monoid homomorphism from Σ^* to $(d \times d)$-matrices with rational entries, we are interested in the computability of the algebraic closure of $\{\varphi(w) \mid w \in L(G)\}$.

Theorem 14. Given an indexed grammar G over an alphabet Σ and a monoid morphism $\varphi : \Sigma^* \to \mathrm{M}_d(\mathbb{Q})$, the algebraic closure of $\varphi(L(G))$ is not computable.

The proof is by reduction from boundedness problem for reset vector addition systems with states (reset VASSs), which is known to be undecidable [14]. Before we proceed with the proof, we give a brief introduction to reset VASSs and the boundedness problem for them.

Reset VASSs. A reset VASS $\mathcal{V}$ with dimension $n \in \mathbb{N}$ is a tuple (Q, q_0, Δ) where Q is a finite set of states, $q_0 \in Q$ is the initial state and $\Delta \subseteq Q \times \{+1, 0, -1, \mathrm{reset}\}^n \times Q$ is the transition relation. A run of $\mathcal{V}$ is a sequence $(q_0, \boldsymbol{x}_0), (q_1, \boldsymbol{x}_1), \cdots, (q_k, \boldsymbol{x}_k)$ of state-vectors in $Q \times \mathbb{Z}^n_{\geq 0}$ such that for all $j \in \{1, \ldots, k\}$, there exists $(p, \boldsymbol{v}, q) \in \Delta$ satisfying the following

- $q_{j-1} = p$, $q_j = q$, and $\boldsymbol{x}_0 = \mathbf{0}$,
- the ith coordinate of $\boldsymbol{x}_j$ is zero if the corresponding coordinate in $\boldsymbol{v}$ is reset, and otherwise it is the summation of the ith coordinate of $\boldsymbol{v}$ and $\boldsymbol{x}_{j-1}$.

A configuration $(q, \boldsymbol{x}) \in Q \times \mathbb{Z}^n_{\geq 0}$ is reachable if for some $k \in \mathbb{N}$ there is a run $(q_0, \boldsymbol{x}_0), (q_1, \boldsymbol{x}_2), \cdots, (q_k, \boldsymbol{x}_k)$ with $q_k = q$ and $\boldsymbol{x}_k = \boldsymbol{x}$. Denote by $\mathrm{Reach}(\mathcal{V})$ the set of reachable configurations of $\mathcal{V}$ in $Q \times \mathbb{Z}^n_{\geq 0}$. The Boundedness problem asks, given a reset VASS $\mathcal{V}$, whether $\mathrm{Reach}(\mathcal{V})$ is finite.

Matrix Identities and Outer Product. Recall that the outer product of two column vector $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{Q}^{n\times 1}$, denoted by $\boldsymbol{x} \otimes \boldsymbol{z}$, is defined by $(\boldsymbol{x} \otimes \boldsymbol{z})_{i,j} := x_i y_j$ for all $1 \le i, j \le n$.

Below, we denote by $\mathbf{0} \in \mathbb{Q}^{(n+1)\times 1}$ and $\mathbf{1} \in \mathbb{Q}^{(n+1)\times 1}$ the all-zero and all-one column vectors, respectively. Denote by column vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{n+1} \in \mathbb{Q}^{(n+1)\times 1}$ the standard basis. Fix a column vector $\boldsymbol{y} = (yx_1, \ldots, yx_n, y)^\top$ where $x_1, \ldots, x_n, y \in \mathbb{Q}$.

Fact 15. The outer product $\boldsymbol{y} \otimes \mathbf{1}$ is a matrix with identical columns equal to $\boldsymbol{y}$:

$$\boldsymbol{y} \otimes \mathbf{1} = \boldsymbol{y}\mathbf{1}^\top = \begin{pmatrix} yx_1 & yx_1 & \ldots & yx_1 \\ yx_2 & yx_2 & \ldots & yx_2 \\ \vdots & \vdots & \ddots & \vdots \\ yx_n & yx_n & \ldots & yx_n \\ y & y & \ldots & y \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{10}$$

Similarly, $\mathbf{1} \otimes \boldsymbol{y}$ is a matrix with repeated rows $\boldsymbol{y}^\top$. The product $\mathbf{1} \otimes \mathbf{1}$ is a matrix whose every entry is 1.

The following claims are essential for the reduction:

Claim 15.1. Let $k \in \{1, \ldots, n\}$. The product $(\boldsymbol{y} \otimes \mathbf{1})(\mathbf{1} \otimes (\boldsymbol{e}_k + \boldsymbol{e}_{n+1}))(\boldsymbol{y} \otimes \mathbf{1})$ is a matrix whose columns are all equal to $(n+1)y(x_k+1)\boldsymbol{y}$.

Proof of Claim. First we observe that the following product L is a matrix where all columns are zero except the k-th and $(n+1)$-th column being $(n+1)\boldsymbol{y}$:

$$\begin{aligned} L = (\boldsymbol{y} \otimes \mathbf{1})(\mathbf{1} \otimes (\boldsymbol{e}_k + \boldsymbol{e}_{n+1})) &= (\boldsymbol{y}\mathbf{1}^\top)(\mathbf{1}(\boldsymbol{e}_k + \boldsymbol{e}_{n+1})^\top) \\ &= \boldsymbol{y}(\mathbf{1}^\top\mathbf{1})(\boldsymbol{e}_k + \boldsymbol{e}_{n+1})^\top \\ &= (n+1)\boldsymbol{y} \otimes (\boldsymbol{e}_k + \boldsymbol{e}_{n+1}) \qquad \text{as } \mathbf{1}^\top\mathbf{1} = n+1\,. \end{aligned}$$

Recall that all columns of $R = (\boldsymbol{y} \otimes \mathbf{1})$ are equal to $\boldsymbol{y}$. Therefore

$$\begin{aligned} LR &= (n+1)\begin{pmatrix} 0 & \ldots & 0 & yx_1 & 0 & \ldots & 0 & yx_1 \\ 0 & \ldots & 0 & yx_2 & 0 & \ldots & 0 & yx_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & yx_n & 0 & \ldots & 0 & yx_n \\ 0 & \ldots & 0 & y & 0 & \ldots & 0 & y \end{pmatrix}\begin{pmatrix} yx_1 & yx_1 & \ldots & yx_1 \\ yx_2 & yx_2 & \ldots & yx_2 \\ \vdots & \vdots & \ddots & \vdots \\ yx_n & yx_n & \ldots & yx_n \\ y & y & \ldots & y \end{pmatrix} \\ &= (n+1)\begin{pmatrix} yx_1(yx_k+y) & \ldots & yx_1(yx_k+y) \\ yx_2(yx_k+y) & \ldots & yx_2(yx_k+y) \\ \vdots & \ddots & \vdots \\ yx_n(yx_k+y) & \ldots & yx_n(yx_k+y) \\ y(yx_k+y) & \ldots & y(yx_k+y) \end{pmatrix}. \end{aligned}$$

We get that all columns of LR are $(n+1)y(x_k+1)\boldsymbol{y}$.

♣

Given a matrix M, we denote by $M_{i:}$ its i-th row. The proof of this second claim is provided in Section F.

Claim 15.2. Let $M \in \mathbb{Q}^{(n+1)\times(n+1)}$. Then

$$(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = \begin{cases} \mathbf{0}^\top & \text{if } M_{i:} = \mathbf{0}^\top, \\ (yx_i \pm y)\mathbf{1}^\top & \text{if } M_{i:} = (\boldsymbol{e}_i \pm \boldsymbol{e}_{n+1})^\top, \\ (yx_i)\mathbf{1}^\top & \text{if } M_{i:} = \boldsymbol{e}_i^\top. \end{cases}$$

Now we are in a position to formalize the reduction:

Proof of Theorem 14. Given a reset VASS (Q, q_0, Δ), we construct an index grammar G over a finite alphabet Σ and $\varphi : \Sigma^* \to \mathrm{M}_d(\mathbb{Q})$ a monoid homomorphism from Σ^* to $(d \times d)$-matrices with rational entries such that

$$\overline{\varphi(L(G))} \text{ has dimension at most one} \iff \mathrm{Reach}(\mathcal{V}) \text{ is finite.}$$

In our reduction, without loss of generality, we assume that $\mathcal{V}$ is such that, for all transitions $(p, \boldsymbol{v}, q) \in \Delta$, the vector $\boldsymbol{v}$ has at most one negative coordinate (as otherwise, introduction of more new states can convert $\mathcal{V}$ to satisfy this assumption). Roughly speaking, the reduction is such that for each sequence of transitions $\delta_1, \ldots, \delta_k$ of $\mathcal{V}$ the grammar G produces a word over the alphabet $\{\delta, t_\delta : \delta \in \Delta\} \cup \{\delta_0\}$ as follows

$$w = w_k\, t_{\delta_k}\, w_k\, t_{\delta_{k-1}}\, w_{k-1}\, t_\delta k - 2\, w_{k-2} \cdots w_3\, t_{\delta_2}\, w_2\, t_{\delta_1} w_1 \tag{11}$$

where w_i is the suffix of $\delta_k\delta_{k-1}\cdots\delta_0$ consisting of $(i+1)$-letter. Observe that $w_i = \delta_i w_{i-1} = \delta_i\delta_{i-1}\cdots\delta_0$ for all $i \in \{k, \ldots, 2\}$. For instance, given the sequence $\delta_1\delta_2\delta_3$ of transitions the grammar produces the following word

$$\underbrace{\delta_3\delta_2\delta_1\delta_0\, t_{\delta_3}\, \delta_3\, \underbrace{\delta_2\delta_1\delta_0\, t_{\delta_2}\, \delta_2\, \underbrace{\delta_1\delta_0\, t_{\delta_1}\, \delta_1\delta_0}_{\text{to check: can } \delta_1 \text{ be taken at } (q_0,\mathbf{0})?}}_{\text{to check: can } \delta_1\delta_2 \text{ be taken at } (q_0,\mathbf{0})?}}_{\text{to check: can } \delta_1\delta_2\delta_3 \text{ be taken at } (q_0,\mathbf{0})?} \tag{12}$$

The idea is that if the sequence of transitions $\delta_1 \cdots \delta_k$ lifts to an invalid run, then $\varphi(w)$ is the zero matrix, and otherwise it is an encoding of the reachable configuration by $\delta_1 \cdots \delta_k$. Similar to the reduction in Proposition 1, we encode the configurations $(q, \boldsymbol{x}) \in Q \times \mathbb{Z}^n_{\geq 0}$ of $\mathcal{V}$ with matrices M consisting of $|Q| \times |Q|$ blocks of $(n+1) \times (n+1)$ matrices. The encoding is such that all blocks of M are zero matrices except possibly one block at index (q, q_0); this indicates that the run began in q_0 and reached q, with δ_k being the last transition (note that this is the reverse block compared to Proposition 1, where such a block has index (q_0, q)). Given a column vector $\boldsymbol{y}$, by Fact 15, the outer product $\boldsymbol{y} \otimes \mathbf{1}$ is a matrix with repeated columns $\boldsymbol{y}$. The vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ of the configuration $(q, \boldsymbol{x})$ is encoded in the nonzero block of M by the matrix $y(\boldsymbol{y} \otimes \mathbf{1})$ where $\boldsymbol{y} = (\boldsymbol{x}, 1)^\top = (x_1, \ldots, x_n, 1)^\top$ is a column vector and y is a nonzero scalar. We may refer to y as the unit of the encoding. We consider the case when the unit is zero, meaning matrices with all blocks zero, encoding of an invalid configuration.

Since the matrix multiplication is associative, we can study $\varphi(w)$ by considering the word backward, as shown in (12). The matrix product $\varphi(\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0)$ aims at checking whether δ_1 lift to a valid run starting at $(q_0, \mathbf{0})$. As briefly explained above, the main idea is that if taking δ_1 is not allowed at the beginning of the run then $\varphi(\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0) = \mathbf{0}$. Otherwise, the construction is such that

$$\varphi(\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0) = c\,\varphi(\delta_1\delta_0)$$

for some nonzero $c \in \mathbb{Q}$, which implies that $\varphi(\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0)$ is a valid encoding of the configuration reached after δ_1. Inductively, given that $\delta_1 \dots \delta_{i-1}$ lifts to a valid run in $\mathcal{V}$, the image of suffix $w_{i-1}t_{\delta_{i-1}}w_{i-2}\cdots t_{\delta_1}w_1$ of w is a multiple of image of w_{i-1}:

$$\varphi(w_{i-1}t_{\delta_{i-1}}w_{i-2}\cdots t_{\delta_1}w_1) = c'\,\varphi(w_{i-1})$$

for some nonzero $c' \in \mathbb{Q}$. The construction next checks whether δ_i can be taken in $\mathcal{V}$ after $\delta_1 \dots \delta_{i-1}$. This task is performed by the letter t_{δ_i} in the the infix $w_i t_{\delta_i}$ right before the suffix $w_{i-1}t_{\delta_{i-1}}w_{i-2}\cdots t_{\delta_1}w_1$. By the invariant, on the right of $\varphi(t_{\delta_i})$ we have $c\,\varphi(\delta_i\delta_{i-1}\cdots\delta_0)$ and on the left we have $\varphi(\delta_i\delta_{i-1}\cdots\delta_0)$, both encoding of the configuration reached after taking $\delta_1 \dots \delta_i$, but possibly with different units. These two copies enable the simulation of a quadratic update to the vector values and allow detection of whether any coordinate becomes -1 as a result of δ_i. For instance, let $\boldsymbol{x} = (x_1, \dots, x_n)$ be the vector reached after taking $\delta_0\delta_1 \dots \delta_i$, the two encoding are in the form of $y_1((\boldsymbol{x}, 1) \otimes \mathbf{1})$ and $y_2((\boldsymbol{x}, 1) \otimes \mathbf{1})$ for some nonzero scalars y_1, y_2. Let $\delta_i = (p, \boldsymbol{v}, q)$ be such that the ℓ-th coordinate of $\boldsymbol{v}$ is -1 (the only coordinate by assumption). By Claim 15.1, the product of these encoding on the left and right by $\mathbf{1} \otimes (\boldsymbol{e}_\ell + \boldsymbol{e}_{n+1})$ results in a matrix whose columns are all $(n+1)y_1y_2(x_\ell + 1)(\boldsymbol{x}, 1)$. If taking δ_i results the x_ℓ to become -1, this product is zero and gives the encoding of an invalid configuration. Otherwise, it is a valid encoding of $\boldsymbol{x}$ with the unit $(n+1)\,y_1y_2\,(x_\ell + 1)$.

The grammar $G = (V, \Sigma, I, P, S)$ is defined such that

$$V = \{S, A, B, C, D\} \qquad \Sigma = \{\delta, t_\delta \mid \delta \in \Delta\} \cup \{\delta_0\} \qquad I = \{f_\delta \mid \delta \in \Delta\} \cup \{f_{\delta_0}\}$$

where V is the set of variables, I the set of indices, and $S \in V$ the starting variable. We introduce the production rules in P gradually, as it applies during a derivation, showing how it generates words of the form (11). For all $\delta \in \Delta$, the rules

$$S \to Af_{\delta_0} \qquad\qquad A \to Af_\delta$$

generate derivations of the form $S \overset{*}{\Longrightarrow} A\,f_{\delta_k}\,f_{\delta_{k-1}} \dots\, f_{\delta_0}$. The rule $A \to DBC$ can be applied only once along a derivation, generating $S \overset{*}{\Longrightarrow} Df_{\delta_k} \dots f_{\delta_0} Bf_{\delta_k} \dots f_{\delta_0} Cf_{\delta_k} \dots f_{\delta_0}$. Intuitively, the grammar is now committed to simulate the sequence $\delta_1 \cdots \delta_k$ of transitions in $\mathcal{V}$. Next, taking once $Cf_\delta \to BC$, with $\delta \in \Delta$, generates

$$S \overset{*}{\Longrightarrow} Df_{\delta_k} \dots f_{\delta_0} Bf_{\delta_k} \dots f_{\delta_0} Bf_{\delta_{k-1}} \dots f_{\delta_0} Cf_{\delta_{k-1}} \dots f_{\delta_0}\,,$$

and repeating this rule again gives

$$S \overset{*}{\Longrightarrow} Df_{\delta_k} \dots f_{\delta_0} Bf_{\delta_k} \dots f_{\delta_0} Bf_{\delta_{k-1}} \dots f_{\delta_0} Bf_{\delta_{k-2}} \dots Bf_{\delta_1} f_{\delta_0} Bf_{\delta_0} Cf_{\delta_0}.$$

For all $\delta \in \Delta$, the rules

$$Cf_{\delta_0} \to \epsilon \qquad\qquad Bf_\delta \to t_\delta \delta D$$

combined together yields

$$S \overset{*}{\Longrightarrow} Df_{\delta_k} \dots f_{\delta_0}\, t_{\delta_k}\delta_k Df_{\delta_{k-1}} \dots f_{\delta_0}\, t_{\delta_{k-1}}\delta_{k-1} Df_{\delta_{n-2}} \dots f_{\delta_0} \;\dots t_{\delta_2}\delta_2 Df_{\delta_1} f_{\delta_0}\, t_{\delta_1}\delta_1 Df_{\delta_0}\,.$$

Finally, the rules

$$Df_\delta \to \delta D \qquad\qquad D_{\delta_0} \to \delta_0$$

complete the generation of words of the form (11).

It remains to define the morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ where $d = |Q| \times (n+1)$. As described above, each matrix $\varphi(\sigma)$, for all $\sigma \in \Sigma$, consists of $|Q| \times |Q|$ blocks of $(n+1) \times (n+1)$ matrices, with only one nonzero block. Define $\varphi(\delta_0)$ to be such a matrix with the nonzero block $B := (\mathbf{0}, 1) \otimes \mathbf{1}$ at index (q_0, q_0). Observe that $\varphi(\delta_0)$ encodes the initial configuration. Let $\delta \in \Delta$ such that $\delta = (p, \boldsymbol{v}, q)$. The matrix $\varphi(\delta)$ has the nonzero block B at index (q, p) where B is constructed as in Claim 15.2, where for all $i \in \{1, \ldots, n\}$, we define

$$B_{i:} = \begin{cases} \mathbf{0}^\top & \text{if } v_i = \text{reset} \\ \boldsymbol{e}_i^\top & \text{if } v_i = 0 \\ (\boldsymbol{e}_i \pm \boldsymbol{e}_{n+1})^\top & \text{if } v_i = \pm 1 \end{cases}$$

where $B_{i:}$ denotes the i-th row of B. We define $B_{n+1:}$ to be e_{n+1}.

Recall that at most one coordinate of $\boldsymbol{v}$ is -1; let ℓ be the index of this coordinate if it exists, and $\ell = 1$ otherwise. For the letter t_δ, we define $\varphi(t_\delta)$ to be a matrix with the nonzero block B at index (q, q_0), where $B := \mathbf{1} \otimes (\boldsymbol{e}_\ell + \boldsymbol{e}_{n+1})$ is chosen as in Claim 15.2.

The correctness proof follows from the next claim, see the proof in Section F:

Claim 15.3. The set $\text{Reach}(\mathcal{V})$ is finite if, and only if, $\overline{\varphi(L(G))}$ has dimension at most one.

□

Acknowledgements Mahsa Shirmohammadi and Mahsa Naraghi were supported by the ANR grant VeSyAM (ANR-22-CE48-0005). James Worrell and Rida Ait El Manssour were supported by EPSRC Fellowship EP/X033813/1.

References

[1] Alfred V. Aho. Indexed grammars - an extension of context-free grammars. *J. ACM*, 15(4):647–671, 1968.

[2] Jason P. Bell and Daniel Smertnig. Computing the linear hull: Deciding deterministic? and unambiguous? for weighted automata over fields. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023.

[3] Mikolaj Bojanczyk. Factorization forests. In Volker Diekert and Dirk Nowotka, editors, *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, volume 5583 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2009.

[4] Armand Borel. *Linear algebraic groups*, volume 126. Springer Science & Business Media, 2012.

[5] Helmut Boseck. On algebraic and semialgebraic groups and semigroups. In *Seminar Sophus Lie*, volume 3, pages 221–227, 1993.

[6] Georgina Bumpus, Christoph Haase, Stefan Kiefer, Paul-Ioan Stoienescu, and Jonathan Tanner. On the size of finite rational matrix semigroups. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 115:1–115:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[7] Krishnendu Chatterjee, Hongfei Fu, Amir Kafshdar Goharshady, and Ehsan Kafshdar Goharshady. Polynomial invariant generation for non-deterministic recursive programs. In Alastair F. Donaldson and Emina Torlak, editors, *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2020, London, UK, June 15-20, 2020*, pages 672–687. ACM, 2020.

[8] David Cox, John Little, Donal O'shea, and Moss Sweedler. *Ideals, varieties, and algorithms*, volume 3. Springer, 1997.

[9] John Cyphert and Zachary Kincaid. Solvable polynomial ideals: The ideal reflection for program analysis. *Proc. ACM Program. Lang.*, 8(POPL):724–752, 2024.

[10] Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for petri nets is not elementary. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 24–33. ACM, 2019.

[11] Wojciech Czerwinski and Lukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1229–1240. IEEE, 2021.

[12] Wojciech Czerwinski and Georg Zetzsche. An approach to regular separability in vector addition systems. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 341–354. ACM, 2020.

[13] Harm Derksen, Emmanuel Jeandel, and Pascal Koiran. Quantum automata and algebraic groups. *J. Symb. Comput.*, 39(3-4):357–371, 2005.

[14] Catherine Dufourd, Alain Finkel, and Ph Schnoebelen. Reset nets between decidability and undecidability. In *International Colloquium on Automata, Languages, and Programming*, pages 103–115. Springer, 1998.

[15] Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998.

[16] Guillem Godoy and Ashish Tiwari. Invariant checking for programs with procedure calls. In *International Static Analysis Symposium*, pages 326–342. Springer, 2009.

[17] Sumit Gulwani and Ashish Tiwari. Computing procedure summaries for interprocedural analysis. In *European Symposium on Programming*, pages 253–267. Springer, 2007.

[18] John E Hopcroft and Jeffrey D Ullman. Introduction to automata theory, languages and computation. adison-wesley. *Reading, Mass*, 1979.

[19] Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. Polynomial invariants for affine programs. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 530–539. ACM, 2018.

[20] Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. On strongest algebraic program invariants. *J. ACM*, 70(5):29:1–29:22, 2023.

[21] Andreas Humenberger, Maximilian Jaroschek, and Laura Kovács. Invariant generation for multi-path loops with polynomial assignments. In Isil Dillig and Jens Palsberg, editors, *Verification, Model Checking, and Abstract Interpretation - 19th International Conference, VMCAI 2018, Los Angeles, CA, USA, January 7-9, 2018, Proceedings*, volume 10747 of *Lecture Notes in Computer Science*, pages 226–246. Springer, 2018.

[22] James E Humphreys. *Linear algebraic groups*, volume 21. Springer Science & Business Media, 2012.

[23] Gérard Jacob. Un algorithme calculant le cardinal, fini ou infini, des demi-groupes de matrices. *Theor. Comput. Sci.*, 5(2):183–204, 1977.

[24] Zachary Kincaid, John Cyphert, Jason Breck, and Thomas W. Reps. Non-linear reasoning for invariant synthesis. *Proc. ACM Program. Lang.*, 2(POPL):54:1–54:33, 2018.

[25] Jérôme Leroux. The general vector addition system reachability problem by presburger inductive invariants. *Logical Methods in Computer Science*, 6, 2010.

[26] Jérôme Leroux. Vector addition system reachability problem: a short self-contained proof. In *Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 307–316, 2011.

[27] Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021.

[28] Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019.

[29] Arnaldo Mandel and Imre Simon. On finite semigroups of matrices. *Theor. Comput. Sci.*, 5(2):101–111, 1977.

[30] Rida Ait El Manssour, George Kenison, Mahsa Shirmohammadi, and Anton Varonka. Simple linear loops: Algebraic invariants and applications. *Proc. ACM Program. Lang.*, 9(POPL):745–771, 2025.

[31] Markus Müller-Olm, Michael Petter, and Helmut Seidl. Interprocedurally analyzing polynomial identities. In Bruno Durand and Wolfgang Thomas, editors, *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, volume 3884 of *Lecture Notes in Computer Science*, pages 50–67. Springer, 2006.

[32] Markus Müller-Olm and Helmut Seidl. Computing polynomial program invariants. *Inf. Process. Lett.*, 91(5):233–244, 2004.

[33] Markus Müller-Olm and Helmut Seidl. Precise interprocedural analysis through linear algebra. In Neil D. Jones and Xavier Leroy, editors, *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004*, pages 330–341. ACM, 2004.

[34] Klara Nosan, Amaury Pouly, Sylvain Schmitz, Mahsa Shirmohammadi, and James Worrell. On the computation of the Zariski closure of finitely generated groups of matrices. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, pages 129–138, 2022.

[35] Michael S Paterson. Unsolvability in 3× 3 matrices. *Studies in Applied Mathematics*, 49(1):105–107, 1970.

[36] Sriram Sankaranarayanan, Henny Sipma, and Zohar Manna. Non-linear loop invariant generation using Gröbner bases. In Neil D. Jones and Xavier Leroy, editors, *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004*, pages 318–329. ACM, 2004.

[37] Imre Simon. Factorization forests of finite height. *Theor. Comput. Sci.*, 72(1):65–94, 1990.

A Detailed Computations for the Examples in the Overview

We start with Example 1 where $\varphi(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $\varphi(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Note that

$$\varphi(a)^m = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}, \quad \varphi(b)^n = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix},$$

then $\varphi(a)^m\varphi(b)^n = \begin{pmatrix} mn+1 & m \\ n & 1 \end{pmatrix}$. Therefore, the Zariski closure of

$$\varphi(L_C) = \{\varphi(a)^m\varphi(b)^n : m \geq n\}$$

is the zero set of the ideal $I_C = \langle x_{11} - x_{12}x_{21} - 1, x_{22} - 1\rangle$. Moreover, the closure of

$$\varphi(L_R) = \{\varphi(a)^n\varphi(b)^n : n \in \mathbb{N}\}$$

is the zero set of the ideal $I_R = \langle x_{11} - x_{12}x_{21} - 1, x_{22} - 1, x_{12} - x_{21}\rangle$.

For Example 2 where again $\varphi(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $\varphi(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, we have $\varphi(L)$ is stable under multiplication and contains $\{\varphi(a)^n\varphi(b)^n : n \in \mathbb{N}\}$. Therefore,

$$\overline{\{\varphi(a)^{n_1}\varphi(b)^{n_1}\varphi(a)^{n_2}\varphi(b)^{n_2}\varphi(a)^{n_3}\varphi(b)^{n_3} : n_1, n_2, n_3 \in \mathbb{N}\}} \subseteq \overline{\varphi(L)}.$$

We will also show that

$$\overline{\varphi(L)} \subseteq \overline{\{\varphi(a)^{n_1}\varphi(b)^{n_1}\varphi(a)^{n_2}\varphi(b)^{n_2}\varphi(a)^{n_3}\varphi(b)^{n_3} : n_1, n_2, n_3 \in \mathbb{N}\}}.$$

In fact, the entries of $\varphi(a)^{n_1}\varphi(b)^{n_1}\varphi(a)^{n_2}\varphi(b)^{n_2}\varphi(a)^{n_3}\varphi(b)^{n_3}$ are polynomials in n_1, n_2, n_3. Denote by $M_{ij}(n_1, n_2, n_3)$ the entries of $\varphi(a)^{n_1}\varphi(b)^{n_1}\varphi(a)^{n_2}\varphi(b)^{n_2}\varphi(a)^{n_3}\varphi(b)^{n_3}$ for $1 \leq i, j \leq 2$. We introduce new variables t_1, t_2, t_3 that play the role of the integers n_1, n_2, n_3 and we consider the ideal $J \subseteq \mathbb{Q}[x_{11}, x_{12}, x_{21}, x_{22}, t_1, t_2, t_3]$ defined by

$$J \coloneqq \langle x_{11} - M_{11}(t_1, t_2, t_3), x_{12} - M_{12}(t_1, t_2, t_3), x_{21} - M_{21}(t_1, t_2, t_3), x_{22} - M_{22}(t_1, t_2, t_3)\rangle.$$

By eliminating the variables t_1, t_2, t_3 with the help of a computer algebra system such as Macaulay2 we find that these entries satisfy only one equation $x_{11}x_{22} - x_{12}x_{21} - 1 = 0$. Since the matrices $\varphi(a)$ and $\varphi(b)$ have determinant 1, every element in $\varphi(L)$ has determinant 1. We conclude that $\varphi(L)$ is the zero set of the ideal

$$I_L = \langle x_{11}x_{22} - x_{12}x_{21} - 1\rangle.$$

We now move to Example 3 where we considered two monoid morphisms $\varphi_1 : \Sigma^* \to M_2(\mathbb{Q})$ and $\varphi_2 : \Sigma^* \to M_3(\mathbb{Q})$. We start with φ_1 where $\varphi_1(a) = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$ and $\varphi_1(b) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. To compute the closure of the 1-VASS coverability and reachability we follow the reduction given in the proof of Proposition 1. We define the new morphism $\varphi_1' : \Sigma^* \to M_6(\mathbb{Q})$ as defined in the proof of Proposition 1. Therefore, we have

$$\varphi_1'(a) = \begin{pmatrix} \begin{pmatrix} \varphi_1(a) & 0 \\ 0 & 1 \end{pmatrix} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix} \text{ and } \varphi_1'(b) = \begin{pmatrix} 0_{3\times 3} & \begin{pmatrix} \varphi_1(b) & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} \varphi_1(b) & 0 \\ 0 & 1 \end{pmatrix} & 0_{3\times 3} \end{pmatrix}.$$

Now we solve the COVER CLOSURE and the REACH CLOSURE for the morphism φ_1'. First we define $\mathcal{M} \coloneqq \overline{\langle \varphi_1'(a), \varphi_1'(b) \rangle}$. We claim that the COVER CLOSURE for φ_1' reduces to computing $\mathcal{M}$. In fact, in the coverability language, every non-empty word must start with a. Moreover, for every $w \in \Sigma^*$, since $\omega(a) = 1 > 0$ there exists an integer n_0 large enough that makes every prefix of $a^{n_0}w$ to have a non-negative total weight, which implies $a^{n_0}w$ is in the coverability language. Then, for every $n \geq n_0$, $a^n w$ is in the coverability language. Therefore, for every $n \geq n_0$, $\varphi_1'(a)^n \varphi_1'(w)$ is in the closure of the image of the coverability language by φ_1'. Since the matrix $\varphi_1'(a)$ is stable, that is, $\mathrm{rk}(\varphi_1'(a)) = \mathrm{rk}(\varphi_1'(a)^2)$ the semigroup $\overline{\langle \varphi_1'(a) \rangle}$ is a group and contains an identity element denoted by I_a (See Lemma 10 for more details). Thus, by a similar argument as in the proof of Claim 11.2 which relies on the Zariski continuity we have that $I_a \varphi(w)$ is in the closure of the image of the coverability language by φ_1' for every $w \in \Sigma^*$. We conclude that the closure of the image of the coverability language by φ_1' is $\overline{I_a \mathcal{M}} \cup \{\mathrm{Id}_6\}$.

For the REACH CLOSURE, we first define $\mathcal{N} \coloneqq \overline{\langle \varphi_1'(a)^n \varphi_1'(b)^n, \varphi_1'(b)^n \varphi_1'(a)^n : n \in \mathbb{N} \rangle}$. We claim that the REACH CLOSURE reduces to the computation of $\mathcal{N}$. In fact, every non-empty word in the reachability language must start with a and end with b. Moreover, for every word $w \in \Sigma^*$ such that $\omega(w) = 0$, there exists n_0 large enough that makes every prefix of $a^{n_0} w b^{n_0}$ have a non-negative total weight, which implies $a^{n_0} w b^{n_0}$ is in the reachability language. Then for every $n \geq n_0$, $a^{n_0} w b^{n_0}$ is in the reachability language. Therefore, $\varphi(a)^n \varphi(w) \varphi(b)^n$ is in the closure of the image of the reachability language by φ_1' for every $n \geq n_0$. Since $\varphi_1'(a), \varphi_1'(b)$ are stable matrices, the semigroup $\overline{\langle (\varphi_1'(a), \varphi_1'(b)) \rangle} \subseteq M_6(\overline{\mathbb{Q}}) \times M_6(\overline{\mathbb{Q}})$ is a group and it contains an identity element (I_a, I_b) (See Lemma 10). Then by the Zariski continuity and similar argument used in the proof of Claim 12.2 we have that $I_a \varphi(w) I_b$ is in the closure of the image of the reachability language by φ_1' for every $w \in \Sigma^*$ with $\omega(w) = 0$. Moreover, for every word $w \in \Sigma^*$ such that $\omega(w)$, we have $\varphi_1'(w) \in \mathcal{N}$ (this is also a consequence of the stability of the matrices $\varphi_1'(a), \varphi_1'(b)$ and the Zariski continuity). Thus we conclude that the closure of the image of the reachability language by φ_1' is $\overline{I_a \mathcal{N} I_b} \cup \{\mathrm{Id}_6\}$.

Let us now compute $\mathcal{M}$, note that

$$\varphi_1(a)^m \varphi_1(b)^n = \begin{pmatrix} 2^m & 0 \\ n4^m & 4^m \end{pmatrix}, \quad \varphi_1(b)^n \varphi_1(a)^m = \begin{pmatrix} 2^m & 0 \\ n2^m & 4^m \end{pmatrix},$$

then, define $M(x,y) \coloneqq \begin{pmatrix} x & 0 & 0 \\ y & x^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ for every $x, y \in \overline{\mathbb{Q}}$. Hence,

$$\mathcal{M} = \left\{ \begin{pmatrix} M(x,y) & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & M(x,y) \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ M(x,y) & 0_{3\times 3} \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & M(x,y) \end{pmatrix}, \right.$$
$$\left. \begin{pmatrix} M(1,y) & 0_{3\times 3} \\ 0_{3\times 3} & M(1,y) \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & M(1,y) \\ M(1,y) & 0_{3\times 3} \end{pmatrix} : x, y \in \overline{\mathbb{Q}} \right\},$$

where the matrices $\begin{pmatrix} M(1,y) & 0_{3\times 3} \\ 0_{3\times 3} & M(1,y) \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & M(1,y) \\ M(1,y) & 0_{3\times 3} \end{pmatrix}$ correspond to $\overline{\langle \varphi_1'(b) \rangle}$.

- The closure of the image of the coverability language by φ_1' is

$$\overline{I_a \mathcal{M}} \cup \{\mathrm{Id}_6\} = \left\{ \mathrm{Id}_6, \begin{pmatrix} M(x,y) & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & M(x,y) \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix} : x, y \in \overline{\mathbb{Q}} \right\}.$$

- Note that, the Zariski closure $\overline{\langle \varphi_1(a)^n \varphi_1(b)^n : n \in \mathbb{N} \rangle}$ is $\left\{ \begin{pmatrix} x & 0 \\ y & x^2 \end{pmatrix} : x, y \in \overline{\mathbb{Q}} \right\}$, which implies

that in this example we have $\mathcal{M}$, and $\mathcal{N}$ coinside, that is,

$$\mathcal{N} \coloneqq \overline{\langle \varphi_1'(a)^n \varphi_1'(b)^n, \varphi_1'(b)^n \varphi_1'(a)^n : n \in \mathbb{N} \rangle} = \overline{\langle \varphi_1'(a), \varphi_1'(b) \rangle} = \mathcal{M}.$$

Thus, the closure of the image of the reachability language by φ_1' is

$$\overline{I_a \mathcal{M} I_b} \cup \{\mathrm{Id}_6\} = \left\{ \mathrm{Id}_6, \begin{pmatrix} M(x,y) & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}, \begin{pmatrix} 0_{3\times 3} & M(x,y) \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix} : x, y \in \overline{\mathbb{Q}} \right\}.$$

We go back to the original problem of computing the closure of the 1-VASS coverability and reachability for the morphism φ_1. As explained in the proof of Proposition 1 we conclude that the closures of $\varphi_1(L_C)$ and $\varphi_1(L_R)$ coincide and are given as the zero set of the ideal $\langle x_{12}, x_{11}^2 - x_{22} \rangle$.

Now we consider the morphism φ_2 where

$$\varphi_2(a) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \varphi_2(b) = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

We repeat the same steps as we have done for the morphism φ_1'. We define again the new morphism $\varphi_2' : \Sigma^* \to M_8(\mathbb{Q})$ as in the proof of Proposition 1:

$$\varphi_2'(a) = \begin{pmatrix} \begin{pmatrix} \varphi_2(a) & 0 \\ 0 & 1 \end{pmatrix} & 0_{4\times 4} \\ 0_{4\times 4} & 0_{4\times 4} \end{pmatrix}, \quad \varphi_2'(b) = \begin{pmatrix} 0_{4\times 4} & \begin{pmatrix} \varphi_2(b) & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} \varphi_2(b) & 0 \\ 0 & 1 \end{pmatrix} & 0_{4\times 4} \end{pmatrix}.$$

Note that:

$$\varphi_2(a)^m \varphi_2(b)^n = \begin{pmatrix} 1 & m-n & 1/2(m^2 - n^2 + n - m + mn) \\ 0 & 1 & m+n \\ 0 & 0 & 1 \end{pmatrix}.$$

In fact, the entry x_{12} reflects the number of $\varphi_2(a)$ minus the number of $\varphi_2(b)$ in every product of $\varphi_2(a)$'s and $\varphi_2(b)$'s. Thus, we have

$$\begin{aligned} \mathcal{M} &\coloneqq \overline{\langle \varphi_2'(a), \varphi_2'(b) \rangle} \\ &= \left\{ \begin{pmatrix} M(x,y,z) & 0_{4\times 4} \\ 0_{4\times 4} & 0_{4\times 4} \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & M(x,y,z) \\ 0_{4\times 4} & 0_{4\times 4} \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & 0_{4\times 4} \\ M(x,y,z) & 0_{4\times 4} \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & 0_{4\times 4} \\ 0_{4\times 4} & M(x,y,z) \end{pmatrix}, \right. \\ &\qquad \left. \begin{pmatrix} M(-x,x,\frac{x(x-1)}{2}) & 0_{4\times 4} \\ 0_{4\times 4} & M(-x,x,\frac{x(x-1)}{2}) \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & M(-x,x,\frac{x(x-1)}{2}) \\ M(-x,x,\frac{x(x-1)}{2}) & 0_{4\times 4} \end{pmatrix} : x, y, z \in \overline{\mathbb{Q}} \right\}, \end{aligned}$$

where $M(x,y,z) \coloneqq \begin{pmatrix} 1 & x & z & 0 \\ 0 & 1 & y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$. Then the closure of the image of the coverability language by φ_2' is

$$\begin{aligned} \overline{I_a \mathcal{M}} \cup \{\mathrm{Id}_8\} = \left\{ \mathrm{Id}_8, \right. &\begin{pmatrix} M(x,y,z) & 0_{4\times 4} \\ 0_{4\times 4} & 0_{4\times 4} \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & M(x,y,z) \\ 0_{4\times 4} & 0_{4\times 4} \end{pmatrix}, \begin{pmatrix} 0_{4\times 4} & 0_{4\times 4} \\ M(x,y,z) & 0_{4\times 4} \end{pmatrix}, \\ &\left. \begin{pmatrix} 0_{4\times 4} & 0_{4\times 4} \\ 0_{4\times 4} & M(x,y,z) \end{pmatrix} : x, y, z \in \overline{\mathbb{Q}} \right\}. \end{aligned} \tag{13}$$

In order to compute the closure of the image of the reachability language by φ_2', we first need to compute

$$\mathcal{N} \coloneqq \overline{\langle \varphi_2'(a)^n \varphi_2'(b)^n, \varphi_2'(b)^n \varphi_2'(a)^n : n \in \mathbb{N} \rangle}.$$

By computing $\varphi_2'(a)^n \varphi_2'(b)^n, \varphi_2'(b)^n \varphi_2'(a)^n$ for any $n \in \mathbb{N}$ we can realize that:

$$\mathcal{N} = \left\{ \begin{pmatrix} M(0,y,z) & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{pmatrix}, \begin{pmatrix} 0_{4\times4} & M(0,y,z) \\ 0_{4\times4} & 0_{4\times4} \end{pmatrix}, \begin{pmatrix} 0_{4\times4} & 0_{4\times4} \\ M(0,y,z) & 0_{4\times4} \end{pmatrix}, \begin{pmatrix} 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & M(0,y,z) \end{pmatrix} : y, z \in \overline{\mathbb{Q}} \right\}.$$

Hence, the closure of the image of the reachability language by φ_2' is:

$$\overline{I_a \mathcal{N} I_b} \cup \{\mathrm{Id}_8\} = \left\{ \mathrm{Id}_8, \begin{pmatrix} M(0,y,z) & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{pmatrix}, \begin{pmatrix} 0_{4\times4} & M(0,y,z) \\ 0_{4\times4} & 0_{4\times4} \end{pmatrix} : y, z \in \overline{\mathbb{Q}} \right\}. \tag{14}$$

From Equation (14) and Equation (13) we conclude that the closures of $\varphi_2(L_C)$ and $\varphi_2(L_R)$ are given as the zero set of the ideals $I_C = \langle x_{11} - 1, x_{22} - 1, x_{33} - 1, x_{21}, x_{31}, x_{32} \rangle$ and $I_R = I_C + \langle x_{12} \rangle$ respectively.

B Proof of Proposition 1

Proposition 1. The problem of computing the Zariski closure of a 1-VASS coverability language is interreducible with Cover Closure. The problem of computing the Zariski closure of a 1-VASS reachability language is interreducible with Reach Closure.

Proof. We give the proof for the case of 1-VASS coverability languages. The case of reachability languages follows with minor modifications. Since the language $L_C(\omega)$ is the coverability language of a (one-state) 1-VASS, it suffices to reduce the problem of computing the closure of a 1-VASS coverability language to Cover Closure. To this end, we first observe that every 1-VASS coverability language is the homomorphic image of a language of the form $L \cap L_C(\omega) \subseteq \Sigma^*$ for some finite alphabet Σ and regular language L. Indeed, given a 1-VASS $\mathcal{V}$, let Σ be the set of transitions of $\mathcal{V}$, L the set of sequences of transitions from an initial to accepting state, and ω the map assigning each transition to its weight. Then the coverability language of $\mathcal{V}$ is the image of $L \cap L_C(\omega)$ under the morphism that maps each transition to its label. To realize the claimed reduction, it thus suffices to reduce the problem of computing the Zariski closure of the set $\varphi(L \cap L_C)$ for a morphism $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ to Cover Closure. This is shown in the following claim:

Claim 15.1. The problem of computing the Zariski closure of $\varphi(L_C(\omega) \cap L)$, given a regular language $L \subseteq \Sigma^*$ and morphisms $\varphi : \Sigma^* \to M_d(\mathbb{Q})$ and $\omega : \Sigma^* \to \mathbb{Z}$, can be reduced to Cover Closure problem.

Proof of Claim. Let $\mathcal{A}$ be a deterministic finite automaton that accepts language L. Suppose that A has set of states $Q = \{q_1, \ldots, q_k\}$ with q_1 initial and $F \subseteq Q$ the set of accepting states. Define a morphism $\varphi' : \Sigma^* \to M_{k(d+1)}(\mathbb{Q})$ where each $\varphi'(\sigma)$ is a block matrix of the form

$$\varphi'(\sigma) \coloneqq \begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,k} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ M_{k,1} & M_{k,2} & \cdots & M_{k,k} \end{pmatrix},$$

where blocks $M_{i,j}$ are zero except for those corresponding to transitions (q_i, σ, q_j) of $\mathcal{A}$ labeled by σ. In such cases, for each transition from state q_i to q_j, we set $M_{i,j} = \begin{pmatrix} \varphi(\sigma) & 0 \\ 0 & 1 \end{pmatrix}$.[2] Then the Zariski closure of $\varphi(L_C(\omega) \cap L)$ can be computed from that of $\varphi'(L_C(\omega))$ as follows. First intersect $\varphi'(L_C(\omega))$ with the (closed) set of matrices M such that for some state $q_i \in F$ the bottom-right entry of block M_{1i} equals 1. Then map the resulting set into $M_d(\mathbb{Q})$ by sending matrix M to $\sum_{q_i \in F} M_{1i}$ and projecting out the last row and column. ♣

Having established the claim we have completed the proof of the proposition. □

Corollary 1. The problems Cover Closure and Reach Closure can be reduced to the special case in which each alphabet symbol $\sigma \in \Sigma$ has weight $\omega(\sigma) \in \{-1, 0, 1\}$.

Proof. Every 1-VASS coverability and reachability language can be realized by 1-VASS in which the weight of each transition lies in $\{-1, 0, 1\}$ (general transitions can be handled by inserting intermediate states and ε-transitions). Applying the reduction in the proof of Proposition 1 to such a 1-VASS leads to instances of Cover Closure and Reach Closure in which the weight map $\omega : \Sigma^* \to \mathbb{Z}$ satisfies $\omega(\sigma) \in \{-1, 0, +1\}$ for all $\sigma \in \Sigma$. □

C Background on Exterior Algebra

Let $V = \mathbb{Q}^d$ be the vector space over the field $\mathbb{Q}$, its *exterior algebra* ΛV is a graded $\mathbb{Q}$-vector space equipped with an associative, bilinear, and antisymmetric wedge product $\wedge : \Lambda V \times \Lambda V \to \Lambda V$. It decomposes as

$$\Lambda V = \bigoplus_{r=0}^{n} \Lambda^r V,$$

where $\Lambda^r V$ is the r^{th} exterior power, spanned by wedge product $v_1 \wedge \cdots \wedge v_r$ with $v_i \in V$. Let $e_1, \ldots, e_d$ denotes the standard basis of $\mathbb{Q}^d$, then a basis of $\Lambda^r V$ is given by $e_{i_1} \wedge \cdots \wedge e_{i_r}$ with $1 \leq i_1 < \cdots < i_r \leq d$, and $\dim \Lambda^r V = \binom{d}{r}$ (with $\binom{d}{r} = 0$ for $r > d$). Moreover, a wedge product $v_1 \wedge \cdots \wedge v_r$ is nonzero if and only if the vectors $v_1, \ldots, v_r$ are linearly independent.

The Grassmanian $\mathrm{Gr}(r, d)$ is the set of all r-dimensional subspaces of $\mathbb{Q}^d$.There is an injective map

$$\iota : \mathrm{Gr}(r, d) \to \mathbb{P}(\Lambda^r V)$$

defined by mapping a subspace $W \in \mathrm{Gr}(r, d)$ with basis $v_1, \ldots, v_r$ to $\iota(W) = [v_1 \wedge \cdots \wedge v_r]$. The map is well-defined since if $v_1 \wedge \cdots \wedge v_r = \alpha u_1 \wedge \cdots \wedge u_r$ for some scalar $\alpha \in \mathbb{Q} \setminus \{0\}$, so $\iota(W)$ is defined up to scalar multiplication.

For subspaces $W_1, W_2 \subseteq \mathbb{Q}^d$, we have $W_1 \cap W_2 = \{0\}$ if and only if $\iota(W_1) \wedge \iota(W_2) \neq 0$.

D Missing Proofs of Section 3

Claim 7.1. Let $M_1, \ldots, M_m \in M_d(\mathbb{Q})$ be a rank-r sequence. Then $\mathrm{rk}(M_i \cdots M_k) = r$ for all $1 \leq i < k \leq m$. Furthermore $\mathrm{im}(M_{k+1}) \cap \ker(M_k) = \{0\}$ for $k \neq m$.

[2] The bottom-right entry 1 of the matrices $M_{i,j}$ in $\varphi'(\sigma)$ allows us to distinguish between the presence of a transition (q_i, σ, q_j) with $\varphi(\sigma) = \mathbf{0}$ (modeled by $\begin{pmatrix} \mathbf{0} & 0 \\ 0 & 1 \end{pmatrix}$) and the absence of a transition (q_i, σ, q_j) (modeled by setting the whole block $M_{i,j}$ to be zero). Note that this corrects an oversight in [21, Section 4].

Proof of Claim. For an arbitrary $i \leq k$, the product $M_1 \cdots M_m$ can be divided as

$$M_1 \cdots M_m = (M_1 \cdots M_{i-1})(M_i \cdots M_k)(M_{k+1} \cdots M_m).$$

Since $\text{rank}(M_1 \cdots M_m) = r \leq \text{rank}(M_i \cdots M_k) \leq \text{rank}(M_i) = r$, all segments of the product must have rank r.

To prove the second part, observe the following identity for any pair of matrices $A, B \in M_{d\times d}(\mathbb{Q})$:

$$\text{rk}(AB) = \text{rk}(B) - \dim(\ker(A) \cap \text{im}(B)).$$

Applying this to $M_k M_{k+1}$, and using that each M_i has rank r, we have:

$$r = \text{rk}(M_k M_{k+1}) = \text{rk}(M_{k+1}) - \dim(\ker(M_k) \cap \text{im}(M_{k+1})).$$

Hence, $\dim(\text{im}(M_{k+1}) \cap \ker(M_k)) = 0$, and therefore $\ker(M_k) \cap \text{im}(M_{k+1}) = \{0\}$.

♣

E Missing Proofs of Section 4

Fact 8. Let $f : X \to Y$ be a continuous function of topological spaces X and Y. Then $\overline{f(S)} = \overline{f(\overline{S})}$ for every subset $S \subseteq X$.

Proof. Since $S \subseteq \overline{S}$, we have $f(S) \subseteq f(\overline{S})$; taking closures gives $\overline{f(S)} \subseteq \overline{f(\overline{S})}$. On the other hand, by definition of continuity we have $f(\overline{S}) \subseteq \overline{f(S)}$. Taking closures gives the reverse inclusion. □

Fact 9. A closed subsemigroup of $GL_d(\overline{\mathbb{Q}})$ is a linear algebraic group.

Proof. Let S be a closed semigroup of $GL_d(\overline{\mathbb{Q}})$. It suffices to argue that S is stable under matrix inversion. Given $M \in S$, clearly $MS \subseteq S$ holds as S is closed under matrix multiplication; moreover, since M is invertible MS is a closed set. Observe that the sequence

$$S \supseteq MS \supseteq M^2S \supseteq \cdots \tag{15}$$

is a decreasing sequence of closed sets. By Noetherian property of ring of polynomial, (15) implies that there exists $k \in \mathbb{N}$ such that $M^k S = M^{k+1}S$. Since M is invertible, the above shows that $S = MS$, implying that $I_n \in MS$. This concludes the proof. □

Lemma 10. Let $M \in M_d(\mathbb{Q})$ be stable. Then $\overline{\{M^n : n > 0\} \cap \{N : \text{rk}(N) = \text{rk}(M)\}}$ is a group.

Proof. Since M is stable we have $\text{im}(M) \cap \ker(M) = \{0\}$. Let $r = \text{rk}(M)$, and let $V = \text{im}(M)$ and $U = \ker(M)$, and choose a basis $B_V = \{e_1, \ldots, e_r\}$ of V and a basis $B_U = \{e_{r+1}, \ldots, e_d\}$ of U. Let $Y \in GL_d(\mathbb{Q})$ be the matrix whose columns are the vectors $e_1, \ldots, e_d$. Then

$$Y^{-1}MY = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}$$

for some $A \in GL_r(\mathbb{Q})$. The powers of M satisfy

$$Y^{-1}M^kY = \begin{pmatrix} A^k & 0 \\ 0 & 0 \end{pmatrix}.$$

Let $\langle A\rangle \subset GL_r(\mathbb{Q})$ be the semigroup generated by A. By Fact 9, $\overline{\langle A\rangle} \cap GL_r(\overline{\mathbb{Q}})$ is a subgroup of $GL_r(\overline{\mathbb{Q}})$. Since the conjugation map $X \mapsto YXY^{-1}$ is a homeomorphism, it commutes with taking closures. Hence,

$$\overline{\langle M\rangle} = Y\left\{\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \;\middle|\; B \in \overline{\langle A\rangle}\right\} Y^{-1}.$$

Intersecting with $\{X \in M_d(\overline{\mathbb{Q}}) \mid \mathrm{rk}(X) = r\}$ restricts B to be invertible:

$$G \coloneqq \overline{\langle M\rangle} \cap \{X \in M_d(\overline{\mathbb{Q}}) \mid \mathrm{rk}(X) = r\} = Y\left\{\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \;\middle|\; B \in \overline{\langle A\rangle} \cap GL_r(\overline{\mathbb{Q}})\right\} Y^{-1}.$$

Thus, since $\overline{\langle A\rangle} \cap GL_r(\overline{\mathbb{Q}})$ is a group, we conclude that G is also a group.

□

F Missing Proofs of Section 7

Claim 15.2. Let $M \in \mathbb{Q}^{(n+1)\times(n+1)}$. Then

$$(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = \begin{cases} \mathbf{0}^\top & \text{if } M_{i:} = \mathbf{0}^\top, \\ (yx_i \pm y)\mathbf{1}^\top & \text{if } M_{i:} = (\boldsymbol{e}_i \pm \boldsymbol{e}_{n+1})^\top, \\ (yx_i)\mathbf{1}^\top & \text{if } M_{i:} = \boldsymbol{e}_i^\top. \end{cases}$$

Proof of Claim. By Fact 15, $\boldsymbol{y}\otimes\mathbf{1}$ is a rank-one matrix with repeated column $\boldsymbol{y} = (yx_1, \ldots, yx_n, y)^\top$. Observe that $(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = (M_{i:}\,\boldsymbol{y})\mathbf{1}^\top$. We now consider each case:

- If $M_{i:} = \mathbf{0}^\top$, then $\mathbf{0}^\top\boldsymbol{y} = 0$. So $(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = \mathbf{0}$.
- If $M_{i:} = (\boldsymbol{e}_i \pm \boldsymbol{e}_{n+1})^\top$, then $M_{i:}\,\boldsymbol{y} = yx_i \pm y$. Hence, $(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = (yx_i \pm y)\mathbf{1}^\top$.
- If $M_{i:} = \boldsymbol{e}_i^\top$, then $M_{i:}\,\boldsymbol{y} = yx_i$, giving $(M(\boldsymbol{y} \otimes \mathbf{1}))_{i:} = (yx_i)\mathbf{1}^\top$.

♣

Claim 15.3. The set $\mathrm{Reach}(\mathcal{V})$ is finite if, and only if, $\overline{\varphi(L(G))}$ has dimension at most one.

Proof of Claim. Recall that given a sequence of transitions $\delta_1, \ldots, \delta_k$ the index grammar produces words of the following form

$$w = w_k\, t_{\delta_k}\, w_k\, t_{\delta_{k-1}}\, w_{k-1}\, t_{\delta_{k-2}}\, w_{k-2} \cdots w_3\, t_{\delta_2}\, w_2\, t_{\delta_1}\, w_1$$

where $w_i = \delta_i w_{i-1} = \delta_i\delta_{i-1}\cdots\delta_0$ for $i \in \{k, \ldots, 2\}$; see (11) and (12). For such sequence of transitions $\delta_1, \ldots, \delta_k$ and the associated word w, we prove the following statement by induction on k:

- if the sequence $(q_0, \mathbf{0}) \xrightarrow{\delta_1} \ldots \xrightarrow{\delta_k} (q_k, \boldsymbol{x}_k)$ does not correspond to a run in $\mathcal{V}$, then $\varphi(w) = \mathbf{0}$. This can happen due to the state mismatch (that is, for some transition δ_i, the source state is not equal to the target state of the preceding transition δ_{i-1}) or a counter value $\boldsymbol{x}_i$ not being nonnegative.
- if the sequence corresponds to a run reaching configuration $(q_k, \boldsymbol{x}_k)$, then the matrix $\varphi(w)$ has exactly one non-zero block with identical columns $y(\boldsymbol{x}_k, 1)^\top$, for some $y \in \mathbb{Q}$.

Base Case ($k = 0$): The run consists of only the initial configuration $(q_0, \mathbf{0})$. The grammar produces the word $w = \delta_0$. By definition, $\varphi(\delta_0)$ is a matrix with a single non-zero block $B = (\mathbf{0}, 1)\otimes\mathbf{1}$ at index (q_0, q_0). This matches the hypothesis for a run where $q_k = q_0, \boldsymbol{x}_k = \mathbf{0}$.

Inductive step ($k > 0$): Assume the hypothesis holds for any sequence of length $k-1$. We now analyze the word w for the sequence $\delta_1, \ldots, \delta_k$, where $\delta_k = (p_k, \boldsymbol{v}, q_k)$. We consider the following distinguished cases for the k-th step. Write w' to be the word assigned to $\delta_1, \ldots, \delta_{k-1}$, *i.e.*

$$w' = w_{k-1}\, t_{\delta_{k-1}}\, w_{k-1}\, t_{\delta_{k-2}}\, w_{k-2} \cdots w_3\, t_{\delta_2}\, w_2\, t_{\delta_1} w_1.$$

- Case 1: A proper prefix of the run is invalid. If the run $\delta_1, \ldots, \delta_{k-1}$ is invalid, then by inductive hypothesis $\varphi(w') = \mathbf{0}$. Since $w = w_k t_{\delta_k} \delta_k w'$, then $\varphi(w) = \varphi(w_k t_{\delta_k} \delta_k)\varphi(w') = \mathbf{0}$, as required.

- Case 2: The run is invalid at step k due to state mismatch. Suppose $\delta_1, \ldots, \delta_{k-1}$ is a run to $(q_{k-1}, \boldsymbol{x}_{k-1})$, but the transition $\delta_k = (p_k, \boldsymbol{v}_k, q_k)$ cannot be applied because its source state p_k does not match the current state q_{k-1}.

  By the inductive hypothesis, $\varphi(w')$ has its single non-zero block at index (q_{k-1}, q_0). The construction of w by the grammar ensures a matrix multiplication of the form $X\varphi(\delta_k)\varphi(w_{k-1})Y$. In this product, the column-state index q_{k-1} of $\varphi(w_{k-1})$ must align with the row-state index p_k of $\varphi(\delta_k)$. Since $p_k \neq q_{k-1}$, this block matrix multiplication results in the $\mathbf{0}$ matrix. Thus, $\varphi(w) = \mathbf{0}$ as required.

- Case 3: The run is invalid at step k due to $\boldsymbol{x}_k$ not being nonneagtive. Suppose the run is valid up to $\delta_1 \ldots \delta_{k-1}$ and leads to $(q_{k-1}, \boldsymbol{x})$, but applying δ_k it becomes invalid, because $\boldsymbol{x} + \boldsymbol{v}_k$ has a negative component in the ℓ-th coordinate. By induction hypothesis, $\varphi(w')$ has exactly one non-zero block $y((\boldsymbol{x}, 1)) \otimes \mathbf{1}$, positioned at index (q_{k-1}, q_0). One can observe that there exists $c_1 \in \mathbb{Q}$, such that $\varphi(\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0) = c_1\varphi(\delta_1\delta_0)$. Therefore

$$\begin{aligned}\varphi(w_2 t_{\delta_2} w_2 t_{\delta_1} w_1) &= \varphi(w_2 t_{\delta_2}\delta_2\delta_1\delta_0 t_{\delta_1}\delta_1\delta_0)\\ &= \varphi(w_2 t_{\delta_2})\varphi(\delta_2)c_1\varphi(\delta_1\delta_0)\\ &= c_1\varphi(w_2)\varphi(t_{\delta_2})\varphi(w_2).\end{aligned}$$

  Inductively, there exists $c \in \mathbb{Q}$ such that

$$\varphi(w) = c\varphi(w_k)\varphi(t_{\delta_k})\varphi(w_k).$$

  Recall that for the letter t_{δ_k}, we define $\varphi(t_{\delta_k})$ to be a matrix with the nonzero block B at index (q, q_0), where $B := \mathbf{1} \otimes (\boldsymbol{e}_\ell + \boldsymbol{e}_{n+1})$. By Claim 15.1, the product $\varphi(w) = c\varphi(w_k)\varphi(t_{\delta_k})\varphi(w_k)$ results in a matrix with exactly one non-zero block whose columns are all $c(n+1)y^2(x_\ell + 1)(\boldsymbol{x}, 1)$. If $x_\ell = -1$, this product is zero.

- Case 4: The run $\delta_1, ..., \delta_k$ is valid. A similar proof to Case 3 establishes the induction statement.

The above induction gives that if $\mathrm{Reach}(\mathcal{V})$ is finite then $\overline{\varphi(L(G))}$ is contained in a finite union of lines and thereby has dimension at most one.

For the reverse direction, assume that $\mathrm{Reach}(\mathcal{V})$ is infinite, we show that $\overline{\varphi(L(G))}$ has dimension strictly more than 1. The proof follows by a similar argument presented in [20, Proof of Proposition 17]. The proof relies on [8, Corollary 4] stating that given an affine variety $V \subseteq \overline{Q^n}$, the dimension of V is equal to the largest integer r for which there exist r variables $x_{i_1}, \ldots, x_{i_r}$ such that the

vanishing ideal $I(V) \cap \mathbb{Q}[x_{i_1}, \ldots, x_{i_r}] = 0$, meaning that $I(V)$ contain no nonzero polynomials in only these variables.

The argument follows by the fact that if $\mathrm{Reach}(\mathcal{V})$ is infinite, there exists some transition δ from state p to q, for some $p, q \in Q$, and some coordinate $i \in \{1, \ldots, n\}$ such that there are infinitely many runs in $\mathcal{V}$ taking δ as the last transition. Hence, there are infinitely many matrices $M \in \varphi(L(G))$ with the nonzero block at index (q, q_0) with growing entries in the i-th row. Recall that all entries in each row are equal and are always a multiple of entries in the $n+1$-th row ((or simply unit). Assume that entries in the i-th row is ℓ, then the unit is larger than 2^ℓ, as after each transition in $\mathcal{V}$ a matrix $\varphi(t_{\delta'})$ was produced to check if the prefix of the run is valid, and this matrix at least doubles the value of the unit. This implies that there is no polynomial relations between the entries of i-th and $(n+1)$-th rows in the (q, q_0)-th block without mentioning other variables in the closure. ♣