Porous invariants for linear systems

Engel Lefaucheux, Joël Ouaknine, David Purser, James Worrell

Formal Methods in System Design
https://doi.org/10.1007/s10703-024-00444-3

Porous invariants for linear systems

Engel Lefaucheux[1] · Joël Ouaknine[2] · David Purser[3] · James Worrell[4]

Received: 31 May 2022 / Accepted: 19 December 2023

Abstract
We introduce the notion of *porous invariants* for multipath affine loops over the integers. These are invariants definable in (fragments of) Presburger arithmetic and, as such, lack certain tame geometrical properties, such a convexity and connectedness. Nevertheless, we show that in many cases such invariants can be automatically synthesised, and moreover can be used to settle reachability questions for various non-trivial classes of affine loops and target sets. For the class of $\mathbb{Z}$-linear invariants (those defined as conjunctions of linear equations with integer coefficients), we show that a strongest such invariant can be computed in polynomial time. For the more general class of $\mathbb{N}$-semi-linear invariants (those defined as Boolean combinations of linear inequalities with integer coefficients), such a strongest invariant need not exist. Here we show that for point targets the existence of a separating invariant is undecidable in general. However we show that such separating invariants can be computed either by restricting the number of program variables or by restricting from multipath to single-path loops. Additionally, we consider porous targets, represented as $\mathbb{Z}$-semi-linear sets (those defined as Boolean combinations of equations with integer coefficients). We show that an invariant can be computed providing the target spans the whole space. We present our tool POROUS, which computes porous invariants.

Keywords Linear dynamical systems · Linear loops · Invariants · Reachability · Presburger arithmetic

1 Introduction

We consider the reachability problem for multipath (or branching) affine loops over the integers, or equivalently for nondeterministic integer linear dynamical systems. A (deterministic) integer linear dynamical system consists of an update matrix $M \in \mathbb{Z}^{d\times d}$ together with an initial point $x^{(0)} \in \mathbb{Z}^d$. We associate to such a system its infinite orbit $(x^{(i)})_{i\in\mathbb{N}}$

✉ Joël Ouaknine
joel@mpi-sws.org

1 Université de Lorraine, Inria, Loria, Nancy, France

2 Max Planck Institute for Software Systems, Saarland Informatics Campus, Saarbrücken, Germany

3 University of Liverpool, Liverpool, UK

4 Department of Computer Science, University of Oxford, Oxford, UK

consisting of the sequence of reachable points defined by the rule $x^{(i+1)} = Mx^{(i)}$. The reachability question then asks, given a target set Y, whether the orbit ever meets Y, i.e., whether there exists some time $i \in \mathbb{N}$ such that $x^{(i)} \in Y$. The nondeterministic reachability question allows the linear update map to be chosen at each step from a fixed finite collection of matrices.

When the orbit does eventually hit the target, one can easily substantiate this by exhibiting the relevant finite prefix. However, establishing non-reachability is intrinsically more difficult, since the orbit consists of an infinite sequence of points. Here one requires a finitary certificate, which must be a relatively simple object that can be inspected and which provides a proof that the set Y is indeed unreachable. Typically, such a certificate will consist of an over-approximation I of the set R of reachable points, in such a manner that one can check both that $Y \cap I = \emptyset$ and $R \subseteq I$; such a set I is called an invariant.

Formally we study the following problem for *inductive invariants*:

The meta problem. *Consider a system defined by an initial vector x and a set of updates, represented by matrices $M_1, \ldots, M_n$. A set I is an inductive invariant of this system if $x^{(0)} \in I$ and $M_i I \subseteq I$ for all $i \in \{1, \ldots, n\}$. Given a target Y, determine whether there exists an inductive invariant I that separates the reachable points of the system from Y, i.e., such that $Y \cap I = \emptyset$.*

The meta problem is parametrised by the type of invariants and targets that are considered; that is, what are the classes of allowable invariant sets I and target sets Y, or equivalently how are such sets allowed to be expressed?

Fixing particular invariant and target domains, a reachability query encounters three possible scenarios: (1) the instance is reachable, (2) the instance is unreachable and a separating invariant from the domain exists, or (3) the instance is unreachable but no separating invariant exists. Ideally, one would wish to provide a sufficiently expressive invariant domain so that the latter case does not occur, whilst keeping the resulting invariants as simple as possible and computable. Unfortunately, it is known that distinguishing reachability (1) from unreachability (2,3) is undecidable in general; and for some invariant domains, within unreachable instances, determining whether a separating invariant exists (i.e., distinguishing (2) from (3)) is undecidable.

We note that the existence of *strongest* inductive invariants is a desirable property for an invariant domain. Given two invariants I and I', we say that I is *stronger* than I' if and only if $I \subseteq I'$; thus *strongest* invariants correspond to *smallest* invariant sets. When strongest invariants exist (and can be computed), separating (2) from (1,3) is easy: compute the strongest invariant, and check whether it excludes the target or not; if so, we are done, and if not, no other invariant (from that class) can possibly work either. However, unless (3) is excluded, computability of the strongest invariant does not necessarily imply that reachability is decidable. Alas, strongest invariants are not always guaranteed to exist for a particular invariant domain, although some separating inductive invariant may still exist for every target (or indeed may not).

In prior work from the literature, typical classes of invariants are usually convex, or finite unions of convex sets. In this paper we consider certain classes of invariants that can have infinitely many 'holes' (albeit in a structured and regular way); we call such sets *porous invariants*. These invariants can be represented via Presburger arithmetic.[1] We shall work instead with the equivalent formulation of semi-linear sets, generalising ultimately periodic sets to higher dimensions, as finite unions of linear sets of the form

[1] Presburger arithmetic is a decidable theory over the natural numbers, comprising Boolean operations, first-order quantification, and addition (but not multiplication).

$\left(b + p_1\mathbb{N} + \cdots + p_m\mathbb{N}\right)$ (by which we mean $\left\{b + a_1p_1 + \cdots + a_mp_m \mid a_1, \ldots, a_m \in \mathbb{N}\right\}$, see Definition 3).

Let us first consider a motivating example:

Example 1 *(Hofstadter's MU Puzzle* [1]*)* Consider the following term-rewriting puzzle over alphabet $\{M, U, I\}$. Start with the word MI, and by applying the following grammar rules (where y and z stand for arbitrary words over our alphabet), we ask whether the word MU can ever be reached.

$$yI \to yIU \quad | \quad My \to Myy \quad | \quad yIIIz \to yUz \quad | \quad yUUz \to yz$$

The answer is *no*. One way to establish this is to keep track of the number of occurrences of the letter 'I' in the words that can be produced, and observe that this number (call it x) will always be congruent to either 1 or 2 modulo 3. In other words, it is not possible to reach the set $\{x \mid x \equiv 0 \mod 3\}$. Indeed, Rules 2 and 3 are the only rules that affect the number of I's, and can be described by the system dynamics $x \mapsto 2x$ and $x \mapsto x - 3$. Hence the MU Puzzle can be viewed as a one-dimensional system with two affine updates,[2] or a two-dimensional system with two linear updates.[3] The set $(1 + 3\mathbb{Z}) \cup (2 + 3\mathbb{Z})$ is an inductive invariant[4], and we wish to automatically synthesise it.

The problem can be rephrased as a safety property of the following multipath loop, verifying that the 'bad' state $x = 0$ is never reached, or equivalently that the loop below can never halt, regardless of the nondeterministic choices made.

```
x := 1
while x ≠ 0
x := 2x || x := x−3      (where || represents nondeterministic branching)
```

The MU Puzzle was presented as a challenge for algorithmic verification in [2]; the tools considered in that paper (and elsewhere, to the best of our knowledge) rely upon the manual provision of an abstract invariant template. Our approach is to find the invariant fully automatically (although one must still abstract from the MU Puzzle the correct formulation as the program $x \mapsto 2x \parallel x \mapsto x - 3$).

Our focus is on the automatic generation of porous invariants for multipath affine loops over the integers, or equivalently nondeterministic integer linear dynamical systems. When we consider affine loops as as linear dynamical systems they do not have loop guards as such. Rather we consider the loop guard as the target of the reachability questions we consider.

- We first consider targets consisting of a single vector (or 'point targets'), and present the classes of invariants and systems for which invariants can and cannot be automatically computed for the reachability question. A summary of the results for linear and semi-linear invariants for these targets is given in Table 1. For completeness we also

[2] One-dimensional affine updates are functions of the form $f(x) = ax + b$.

[3] $\begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} ax + b \\ 1 \end{pmatrix}$ models affine functions using a matrix representation, holding one of the entries fixed to 1.

[4] The stability of this set under our two affine functions is easily checked: both components are invariant under $x \mapsto x - 3$, and $(1 + 3\mathbb{Z}) \mapsto (2 + 6\mathbb{Z}) \subseteq (2 + 3\mathbb{Z})$ under $x \mapsto 2x$, and similarly $(2 + 3\mathbb{Z}) \mapsto (4 + 6\mathbb{Z}) \subseteq (1 + 3\mathbb{Z})$.

Table 1 Results for integer linear dynamical systems for a point target

Dom	D/N	Linear	Semi-linear (SL)
$\mathbb{Z}$	Det	Strongest computable (Theorem 10)	No strongest (Section 4.2)
			Subsumed by $\mathbb{N}$-SL
$\mathbb{Z}$	Non	Strongest computable (Theorem 10)	No strongest (Section 4.2)
$\mathbb{N}$	Det	No strongest (Section 4.2)	No strongest (Section 4.2)
		Subsumed by $\mathbb{N}$-SL	But sufficient computable (Theorem 19)
$\mathbb{N}$	Non	No strongest (Section 4.2)	1d-affine decidable (Theorem 22)
			Undec. in general (Theorem 21)
$\mathbb{R}$	Det	Strongest: affine relations by Karr [4]	Strongest: affine closure on
			Zariski closure (Theorem 8)
$\mathbb{R}$	Non	Strongest: affine relations by Karr [4]	Strongest: affine closure on
			Zariski closure (Theorem 8)
$\mathbb{R}_+$	Det	No strongest (Section 4.2)	No strongest, but sufficient
		Subsumed by $\mathbb{R}_+$-SL	Computable [5]
$\mathbb{R}_+$	Non	No strongest (Section 4.2)	Undecidable [5]

Det/Non refers to deterministic or nondeterministic LDS. "Subsumed by ..." means that sufficient invariants can be generated, but of a more general type

consider $\mathbb{R}, \mathbb{R}_+$-(semi)-linear sets, where we enhance the picture from prior work by showing that strongest $\mathbb{R}$-semi-linear invariants are computable.

- We establish the existence of *strongest* $\mathbb{Z}$-linear invariants, and show that they can be found algorithmically in polynomial time (Theorem 10).
- If a $\mathbb{Z}$-linear invariant is not separating, we may instead look for an $\mathbb{N}$-semi-linear invariant (a class that generalises both $\mathbb{Z}$-semi-linear and $\mathbb{N}$-linear invariants), and we show that such an invariant can always be found for any unreachable point target when dealing with *deterministic* integer linear dynamical systems (Theorem 19).
- However, for nondeterministic integer linear dynamical systems, computing separating $\mathbb{N}$-semi-linear invariants is an undecidable problem in arbitrary dimension (Theorem 21). Nevertheless we show how such invariants can be computed in a low-dimensional setting, in particular for affine updates in one dimension (Theorem 22). As an immediate consequence, this establishes that the multipath loop associated with the MU Puzzle belongs to a class of programs for which we can automatically synthesise $\mathbb{N}$-semi-linear invariants.

- We consider the reachability problem for *porous targets*. That is, where the target is a linear or semi-linear set.
  - For *full-dimensional*[5] $\mathbb{Z}$-linear targets we show that reachability is decidable, and, in the case of unreachability that a $\mathbb{Z}$-semi-linear invariant can always be exhibited as a certificate (Theorem 37). If the target is *not* full-dimensional then the reachability problem is Skolem-hard and undecidable for deterministic and nondeterministic systems respectively.

[5] The affine span covers the entire space.

  - Secondly, we also show that the reachability problem for low-dimensional semi-linear sets is decidable for deterministic LDS (Theorem 40). Note that the Skolem problem is decidable at low orders, so it does not present a barrier in this setting.

- In Sect. 7 we present our tool POROUS which handles one-dimensional affine systems for both point and $\mathbb{Z}$-linear targets, solving both the reachability problem and producing invariants. Inter alia, this allows one to handle the multipath loop derived from the MU Puzzle in fully automated manner.

The present paper extends and strengthens the results of [3]. Firstly, we show that strongest $\mathbb{Z}$-semi-linear invariants can be found in *polynomial time*, whereas [3] merely established decidability. Secondly, we improve the results for *porous targets*, and in particular consider low-dimensional semi-linear targets. Finally, we present all proofs in full.

1.1 Related work

The reachability problem (in arbitrary dimension) for loops with a single affine update, or equivalently for deterministic linear dynamical systems, is decidable in polynomial time for point targets (that is $Y = \{y\}$), as shown by Kannan and Lipton [6]. However for nondeterministic systems (where the update matrix is chosen nondeterministically from a finite set at each time step), reachability was proven undecidable by reduction from the matrix semigroup membership problem [7].

In particular this entails that for unreachable nondeterministic instances we cannot hope to *always* be able to compute a separating invariant. In some cases we may compute the strongest invariant (which may suffice if this invariant happens to be separating for the given reachability query), or we may compute an invariant in sub-cases for which reachability is decidable (for example in low dimensions). For some classes of invariants, it is also undecidable whether an invariant exists (e.g., invariants which are unions of polyhedra [5]).

Various types of invariants have been studied for linear dynamical systems, including polyhedral [5, 8], algebraic [9], and o-minimal [10] invariants. For certain classes of invariants (e.g., algebraic [9]), it is decidable whether a separating invariant exists, notwithstanding the reachability problem being undecidable. Other works (e.g., [11]) use heuristic approaches to generate invariants, without aiming for any sort of completeness.

Kincaid, Breck, Cyphert and Reps [12] study loops with linear updates, examining the closed forms for the variables to prove safety and termination properties. Such closed forms, when expressible in certain arithmetic theories, can be interpreted as another type of invariant and can be used to over-approximate the reachable sets. The work is restricted to a single update function (deterministic loops) and places additional constraints on the updates to bring the closed forms into appropriate theories.

Bozga, Iosif and Konecný's FLATA tool [13] considers affine functions in arbitrary dimension. However, it is restricted to affine functions with finite monoids; in our one-dimensional case this would correspond to limiting oneself to counter-like functions of the form $f(x) = x + b$.

Finkel, Göller and Haase [14], extending Fremont [15], show that reachability in a single dimension is **PSPACE**-complete for polynomial update functions (and allowing states which can be used to control the sequences of updates that can be applied). The affine

functions (and single-state restriction) we consider are a special case, but we focus on producing invariants to disprove reachability.

The reachability problem asks whether there exists a sequence of transitions that reach a given condition. The termination problem asks whether a given condition eventually holds along every possible sequence of transitions. Tools such as AProVE [16] and Büchi Automizer [17] may (dis-)prove reachability in the termination setting, i.e., on *all* branches, but are not suited to asking if a condition can be reached on *some* branch (reachability). Restrictions on the number of switches between the update function can also be considered; [18] shows that reachability is decidable only for a small number of switches.

Inductive invariants specified in Presburger arithmetic have been used to disprove reachability in vector addition systems [19]. A generalisation, the class of 'almost semi-linear sets' [20], also features non-convexity and moreover can capture exactly the reachable points of vector addition systems. Our nondeterministic linear dynamical systems can be seen as vector addition systems over $\mathbb{Z}$ extended with affine updates (rather than only additive updates).

2 Preliminaries

We denote by $\mathbb{Z}$ the set of integers and $\mathbb{N}$ the set of non-negative integers. We say that $x, y \in \mathbb{Z}$ are congruent modulo $d \in \mathbb{N}$, denoted $x \equiv y \mod d$, if d divides $x - y$. Given an integer x and natural d we write $(x \bmod d)$ for the number $y \in \{0, \dots, d-1\}$ such that $y \equiv x \bmod d$.

Definition 2 *(Integer Linear Dynamical Systems)* A d-dimensional integer linear dynamical system (LDS) $(x^{(0)}, \{M_1, \dots, M_k\})$ is defined by an initial point $x^{(0)} \in \mathbb{Z}^d$ and a set of integer matrices $M_1, \dots, M_k \in \mathbb{Z}^{d \times d}$. An LDS is *deterministic* if it comprises a single matrix ($k = 1$) and is otherwise *nondeterministic*.

A point y is *reachable* if there exists $m \in \mathbb{N}$ and $B_1, \dots, B_m$ such that $B_1 \cdots B_m x^{(0)} = y$ and $B_i \in \{M_1, \dots, M_k\}$ for all $1 \le i \le m$.

The *reachability set* $\mathcal{O} \subseteq \mathbb{Z}^d$ of an LDS is the set of reachable points.

The following definition is parameterised by a semiring $\mathbb{K}$, which stands either for $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ or $\mathbb{R}_+$.

Definition 3 *($\mathbb{K}$-semi-linear sets)* A *linear set* L is defined by a base vector $b \in \mathbb{Z}^d$ and period vectors $p_1, \dots, p_k \in \mathbb{Z}^d$ such that

$$L = \{b + a_1 p_1 + \cdots + a_k p_k \mid a_1, \dots, a_k \in \mathbb{K}\}.$$

For convenience we often write $(b + p_1\mathbb{K} + \cdots + p_k\mathbb{K})$ for L. A set is *semi-linear* if it is a finite union of linear sets.

$\mathbb{N}$-semi-linear sets are precisely those definable in Presburger arithmetic($\mathrm{FO}(\mathbb{Z}, +, \le)$) [21]. Likewise, $\mathbb{Z}$-semi-linear sets are those definable in $\mathrm{FO}(\mathbb{Z}, +)$. We also consider their real counterparts, in which the coefficient semiring is either $\mathbb{R}$ or $\mathbb{R}_+$. Note that regardless of the semiring $\mathbb{K}$, the period vectors p_i all lie in $\mathbb{Z}^d$. We say a vector $v \in \mathbb{Z}^d$ is an

admissible direction of a linear set L if adding any $\mathbb{K}$-multiple of v to a point in L is also in L, in particular $L = \left(b + p_1\mathbb{K} + \cdots + p_k\mathbb{K}\right) = \left(b + p_1\mathbb{K} + \cdots + p_k\mathbb{K} + v\mathbb{K}\right)$.

An invariant is simply an overapproximation of the reachability set ($\mathcal{O} \subseteq I$). Typically, we are interested in finding an invariant I that is disjoint from a target, i.e., $I \cap Y = \emptyset$, to show that the orbit $\mathcal{O}$ does not meet Y. We moreover require that the property of being an invariant set be easy to verify. The principal way to do this is to consider inductive invariants:

Definition 4 Given an integer linear dynamical system $(x^{(0)}, \{M_1, \ldots, M_k\})$, a set I is an *inductive invariant* if

- $x^{(0)} \in I$, and
- $\{M_i x \mid x \in I\} \subseteq I$ for all $i \in \{1, \ldots, k\}$.

We are interested in the following problem:

Definition 5 *(Invariant Synthesis Problem)* Given an invariant domain $\mathcal{D}$, an integer linear dynamical system $(x^{(0)}, \{M_1, \ldots, M_k\})$, and a target Y, does there exist an inductive invariant I in $\mathcal{D}$ disjoint from Y?

We foucs on classes $\mathcal{D}$ of inductive invariants that are linear, or semi-linear. When a separating inductive invariant I exists, we also wish to compute it. Since (semi)-linear invariants are enumerable, the computation of invariants can in theory be reduced to the question of their existence; however all of our proofs are constructive.

We also consider the notion of *strongest* invariants, where a strongest invariant is the smallest invariant set I in the prescribed domain that contains $\mathcal{O}$. Such invariants are compelling because they can be used to analyse reachability of any target set in the following sense—either the strongest invariant is separating from the given target, or no invariant in the given domain is separating. Note that strongest invariants do not always exist.

We only consider inductive invariants in the remainder of this paper, and we note when the inductive invariant we compute is also a strongest invariant.

3 $\mathbb{R}$ invariants: $\mathbb{R}$-linear and $\mathbb{R}$-semi-linear

Before delving into porous invariants, let us consider invariants over the real numbers, i.e., $\mathbb{R}$-(semi)-linear sets.

We observe that a strongest $\mathbb{R}$-linear invariant is nothing but the affine hull of the reachability set, which can be computed using Karr's algorithm [4]. Furthermore we show that strongest $\mathbb{R}$-semi-linear invariants also exist and can be computed by combining techniques for computing algebraic invariants [9] and $\mathbb{R}$-linear invariants.

3.1 $\mathbb{R}$-linear invariants

Recall that a set L is $\mathbb{R}$-linear if $L = \left(v_0 + v_1\mathbb{R} + \cdots + v_t\mathbb{R}\right)$ for some $v_0, \ldots, v_t \in \mathbb{Z}^d$ that can be assumed to be linearly independent[6] without loss of generality (and thus $t \leq d$). Given two distinct points of L, every point on the infinite line connecting them must also be in L. Generalising this idea to higher dimensions, given a set $S \subseteq \mathbb{R}^d$, let the affine hull be

$$\mathrm{Aff}(S) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \mid k \in \mathbb{N}, x_i \in S, \lambda_i \in \mathbb{R}, \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

We say the vectors $v_0, \ldots, v_m$ are $\mathbb{Q}$-affinely independent if $v_1 - v_0, \ldots, v_m - v_0$ are $\mathbb{Q}$-linearly independent.

Fix an LDS $(x^{(0)}, \{M_1, \ldots, M_k\})$ and consider its reachability set $\mathcal{O} = \left\{M_{i_m} \cdots M_{i_1} x^{(0)} \mid m \in \mathbb{N}, i_1, \ldots, i_m \in \{1, \ldots, k\}\right\}$. Then $\mathrm{Aff}(\mathcal{O})$ is precisely the strongest $\mathbb{R}$-linear invariant. Karr's algorithm [4, 22] can be used to compute this strongest invariant in polynomial time. The next lemma follows from Theorem 3.1 of [22].

Lemma 6 *Given an LDS $(x^{(0)}, \{M_1, \ldots, M_k\})$ of dimension d, we can compute in time polynomial in d, k, and $\log \mu$ (where $\mu > 0$ is an upper bound on the absolute values of the integers appearing in $x^{(0)}$ and $M_1, \ldots, M_k$), a $\mathbb{Q}$-affinely independent set of integer vectors $R_0 \subseteq \mathcal{O}$ such that:*

1. $x^{(0)} \in R_0$,
2. *the affine span of R_0 and the affine span of $\mathcal{O}$ are the same* ($\mathrm{Aff}(R_0) = \mathrm{Aff}(\mathcal{O})$),
3. *the entries of the vectors in R_0 have absolute value at most $\mu_0 := \mu(d\mu)^d$.*

We highlight that Lemma 6 shows computability of the set R_0 which is a subset of the reachability set (in particular the elements are integer points). This fact will prove useful later in our development of strongest $\mathbb{Z}$-linear invariants in Sect. 4.

Before proving Lemma 6, let us first state a small technical proposition on the growth of matrix powers required in the proof.

Proposition 7 *Let M be a d-dimensional square matrix and x be a vector. Let the maximum entry of M, x have absolute value at most μ. Then the maximal absolute value of an entry of $M^k x$ is at most $d^k \mu^{k+1}$.*

Proof Without loss of generality, assume that the matrix M and vector x consists only of μ. We proceed by induction on k. The base case holds by the assumption that entries of x have absolute value at most μ. The inductive case is as follows:

[6] $v_0, \ldots, v_m$ are $\mathbb{Q}$-linearly independent if there does not exist $a_0, \ldots, a_m \in \mathbb{Q}$, not all 0, such that $a_0 v_0 + \cdots + a_m v_m = 0$.

$$\begin{pmatrix} \mu & \dots & \mu \\ & \ddots & \\ \mu & \dots & \mu \end{pmatrix} \begin{pmatrix} d^{k-1}\mu^k \\ \vdots \\ d^{k-1}\mu^k \end{pmatrix} = \begin{pmatrix} d\mu(d^{k-1}\mu^k) \\ \vdots \\ d\mu(d^{k-1}\mu^k) \end{pmatrix} = \begin{pmatrix} d^k\mu^{(k+1)} \\ \vdots \\ d^k\mu^{(k+1)} \end{pmatrix}$$

□

Proof of Lemma 6 The result of [22, Theorem 3.1] proceeds by finding new points in the reachability set and adding them to a set of points if the new point is linearly independent from the other points of the set. Whilst the result of [22] refers to linear independence, this can be converted to affine independence by increasing the dimension by one.

The procedure works via a pruned version breadth-first search, with nodes only expanded if their children are linearly independent from the current set. Hence, the first point found in the tree is the initial point $x^{(0)}$, and therefore this point is included. The maximum depth of the tree that needs to be explored is d, and so every point included is reached with at most d applications of matrices to $x^{(0)}$. Hence, by Proposition 7, if the largest absolute value of a point or matrix entry is μ, after d iterations, the largest absolute value is $\mu(d\mu)^d$.

The algorithm of [22] runs in polynomial time in the number of arithmetic operations, and we observe that this is also polynomial time in the bit size. The independence checking in the algorithm involves verifying linear independence of at most d vectors all having bit size at most $log(\mu(d\mu)^d) = d\log(d) + (d+1)\log(\mu)$, which can be done in polynomial time in the bit size (for example by the Bareiss algorithm for calculating the determinant).

□

Let $R_0 = \left\{x^{(0)}, r_1, \dots, r_{d'}\right\}$ be obtained as per Lemma 6, with $d' \leq d$. The $\mathbb{R}$-linear invariant of the LDS is the affine span $\mathrm{Aff}(R_0)$, which can be written as the $\mathbb{R}$-linear set $L_0 = \left(x^{(0)} + (r_1 - x^{(0)})\mathbb{R} + \cdots + (r_{d'} - x^{(0)})\mathbb{R}\right)$.

3.2 $\mathbb{R}$-semi-linear invariants

Let us now generalise this approach to $\mathbb{R}$-semi-linear sets, an invariant domain first introduced in [23]. The collection of $\mathbb{R}$-semi-linear sets, $\left\{\bigcup_{i=1}^m L_i \mid m \in \mathbb{N}, L_1, \dots, L_m \text{ are } \mathbb{R}\text{-linear sets}\right\}$, is closed under finite unions and arbitrary intersections.[7] Thus for any given set X, the smallest $\mathbb{R}$-semi-linear set containing X is simply the intersection of all $\mathbb{R}$-semi-linear sets containing X. Let us denote by $\mathrm{SLin}(X)$ the smallest $\mathbb{R}$-semi-linear set that contains X. We are interested in computing $\mathrm{SLin}(\mathcal{O})$:

Theorem 8 *The strongest $\mathbb{R}$-semi-linear invariant $\mathrm{SLin}(\mathcal{O})$ of $\mathcal{O}$ is computable and is inductive.*

First, let us consider the richer class of algebraic sets. Algebraic sets are those that are definable as finite unions and intersections of the zero sets of polynomials. For example, $\{(x, y) \mid xy = 0\}$ describes the union of the lines $x = 0$ and $y = 0$. The (real) Zariski closure $\mathrm{Zar}(X)$ of a set $X \subseteq \mathbb{R}^d$ is the smallest algebraic subset of $\mathbb{R}^d$ containing X. The Zariski

[7] When intersecting a linear set with a semi-linear set, either the latter does not change, or one obtains a finite union of elements of smaller dimension. Thus, in an infinite intersection, only a finite number of intersections affect the original set.

closure of the set of reachable points, $\mathrm{Zar}(\mathcal{O})$, can be computed algorithmically and yields an inductive invariant [9].

An algebraic set A is *irreducible* if whenever $A \subseteq B \cup C$, where B and C are algebraic sets, then we have $A \subseteq B$ or $A \subseteq C$. Any algebraic set can be written effectively as a finite union of irreducible algebraic sets [24].

Proposition 9 *Suppose* $\mathrm{Zar}(X) = A_1 \cup \cdots \cup A_k$, *with* A_i*'s irreducible algebraic sets. Then* $\mathrm{SLin}(X) = \mathrm{Aff}(A_1) \cup \cdots \cup \mathrm{Aff}(A_k)$.

Proof Since semi-linear sets are algebraic we have that $X \subseteq \mathrm{Zar}(X) \subseteq \mathrm{SLin}(X)$ and hence $\mathrm{SLin}(X) \subseteq \mathrm{SLin}(\mathrm{Zar}(X)) \subseteq \mathrm{SLin}(\mathrm{SLin}(X)) = \mathrm{SLin}(X)$. We conclude that $\mathrm{SLin}(X) = \mathrm{SLin}(\mathrm{Zar}(X))$.

Now we have $\mathrm{SLin}(X) \subseteq \mathrm{Aff}(A_1) \cup \cdots \cup \mathrm{Aff}(A_k)$ since the latter is a semi-linear set that contains X. It remains to prove that $\mathrm{Aff}(A_1) \cup \cdots \cup \mathrm{Aff}(A_k) \subseteq \mathrm{SLin}(X)$. For this, write $\mathrm{SLin}(X) = L_1 \cup \cdots \cup L_s$, with the L_j being linear sets. Since each A_i is irreducible and each L_j is algebraic we have that for all i there exists j with $A_i \subseteq L_j$ and hence $\mathrm{Aff}(A_i) \subseteq L_j$. This immediately yields the required inclusion. □

From Proposition 9 we see that $\mathrm{SLin}(\mathcal{O})$ can be obtained by computing $\mathrm{Aff}(A_i)$ for each set A_i arising from the decomposition $\mathrm{Zar}(\mathcal{O}) = A_1 \cup \cdots \cup A_k$ of the Zariski closure of the orbit into irreducible components.[8]

Moreover, the set $\mathrm{SLin}(\mathcal{O})$ is inductive. Indeed, given a matrix M of the LDS and $i \leq k$, for all j, we define the consider set $A_i^j = \{x \in A_i \mid Mx \in A_j\}$, which is clearly algebraic. We have by inductiveness of $\mathrm{Zar}(\mathcal{O})$ that $A_i = \bigcup_j A_i^j$. As A_i is irreducible, one of those sets cannot be a proper subset of A_i. Thus there exists j such that $A_i = A_i^j$ and thus $MA_i \subseteq A_j$. Therefore $M(\mathrm{SLin}(A_i)) = M\mathrm{Aff}(A_i) = \mathrm{Aff}(MA_i) \subseteq \mathrm{Aff}(A_j) \subseteq \mathrm{SLin}(\mathcal{O})$, proving inductiveness.

To complete the proof of Theorem 8 it remains to confirm that affine hulls of algebraic sets can be computed algorithmically. Let us fix an algebraic set A, and let W denote a set variable. Proceed as follows. Start with $W \leftarrow \{x\}$ for some point $x \in A$, and repeatedly make the assignment $W \leftarrow \mathrm{Aff}(W \cup \{y\})$, where $y \in A \backslash W$. Such a point y can always be found using quantifier elimination in the theory of the reals. Each step necessarily increases the dimension, which can occur at most d times, ensuring termination, at which point one has $\mathrm{Aff}(A) = W$.

4 Strongest $\mathbb{Z}$-linear invariants

Recall that a $\mathbb{Z}$-linear set $\left(q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\right)$ is defined by a base vector $q \in \mathbb{Z}^d$ and period vectors $p_1, \ldots, p_n \in \mathbb{Z}^d$. Equivalently, a $\mathbb{Z}$-linear set describes a *lattice*, i.e., $\left(p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\right)$, in d-dimensional space, translated to start from q rather than $\vec{0}$.

We start by showing that the strongest $\mathbb{Z}$-linear invariant can be computed.

[8] While it is convenient to rely on the results of [9], we believe that it is possible and would be more computationally efficient to a give a direct computation of the semi-linear closure that does not go via the Zariski closure.

4.1 Computing the strongest $\mathbb{Z}$-linear invariants

Theorem 10 *Given a d-dimensional dynamical system* $(x^{(0)}, \{M_1, \ldots, M_k\})$, *the strongest* $\mathbb{Z}$*-linear inductive invariant containing the reachability set* $\mathcal{O}$ *exists and can be computed algorithmically in time polynomial in d, k, and* $\log \mu$ (*where* $\mu > 0$ *is an upper bound on the absolute values of the integers appearing in* $x^{(0)}$ *and* $M_1, \ldots, M_k$).

We claim that Algorithm 1 computes the requisite invariant according to Theorem 10. Let us first establish some technical results before proving termination and correctness of the algorithm.

Algorithm 1 Strongest $\mathbb{Z}$-linear invariant for LDS $(x^{(0)}, M_1, \ldots, M_k)$

```
Input x^(0), M_1, ..., M_k
Compute R_0 = {x^(0), r_1, ..., r_d'} ⊆ O according to Lemma 6
L_0 = (x^(0) + (r_1 − x^(0))ℤ + ··· + (r_d' − x^(0))ℤ)
Updated = True
While(Updated):
    Updated = False
    for each M ∈ {M_1, ..., M_k}:
        for each x ∈ R_i:
            x' = Mx
            if x' ∉ L_i:
                R_{i+1} = R_i ∪ {x'}
                L_{i+1} = (x^(0) + Σ_{r∈R_{i+1}} (r − x^(0))ℤ)
                i = i + 1
                Updated = True
return L_i
```

The following proposition asserts that when two points are in a $\mathbb{Z}$-linear set, the direction between these two points can be applied from any point of the set, and hence this direction can be included as a period without altering the set.

Proposition 11 *Let* $L = (q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z})$ *be a* $\mathbb{Z}$*-linear set. If* $x, y \in L$ *then for all* $z \in L$ *and all* $a' \in \mathbb{Z}$ *we have* $z + (y - x)a' \in L$. *In particular, we have* $L = (q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z} + (y - x)\mathbb{Z})$.

Proof If $x = q + a_1p_1 + \cdots + a_np_n$ and $y = q + b_1p_1 + \cdots + b_np_n$ then $y - x = q + b_1p_1 + \cdots + b_np_n - (q + a_1p_1 + \cdots + a_np_n) = (b_1 - a_1)p_1 + \cdots + (b_n - a_n)p_n$.

Then for any $z = q + c_1p_1 + \cdots + c_np_n$, we have $z + a'(y - x) = q + c_1p_1 + \cdots + c_np_n + a'((b_1 - a_1)p_1 + \cdots + (b_n - a_n)p_n)$
$= q + (c_1 + a'(b_1 - a_1))p_1 + \cdots + (c_n + a'(b_n - a_n))p_n)$ where $(c_i + a'(b_i - a_i)) \in \mathbb{Z}$, so $z + a'(y - x) \in L$. □

As a sub-procedure, Algorithm 1 must efficiently decide whether a given point lies in the current candidate invariant L_i.

Proposition 12 *Let $x \in \mathbb{Z}^d$ and $L = \left(x^{(0)} + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\right)$. Suppose μ is an upper bound for the largest absolute value appearing in x and the largest absolute value appearing in all p_i. Then deciding if $x \in L_i$ is in polynomial time in μ, n, d.*

A d-dimensional lattice can always be defined by at most d period vectors. However, our procedure may return a representation containing more than d period vectors.

Example 13 Consider the lattice $((2,2)\mathbb{Z} + (0,6)\mathbb{Z} + (2,6)\mathbb{Z})$, specified with three vectors, which is equivalent to the lattice $((2,0)\mathbb{Z} + (0,2)\mathbb{Z})$. Note that one may not simply pick an independent subset of the periods, as none of the following sets are equal: $((2,2)\mathbb{Z} + (0,6)\mathbb{Z})$, $((2,2)\mathbb{Z} + (2,6)\mathbb{Z})$, $((0,6)\mathbb{Z} + (2,6)\mathbb{Z})$, and $((2,2)\mathbb{Z} + (0,6)\mathbb{Z} + (2,6)\mathbb{Z})$.

The *Hermite normal form* can be used to obtain a basis of the vectors that define the lattice. Consider a lattice $L_i = \left(p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\right)$. The lattice remains the same if p_i is swapped with p_j, if p_i is replaced by $-p_i$, or if p_i is replaced by $p_i + \alpha p_j$ where α is any fixed integer.[9]

The above are the unimodular operations. The Hermite normal form of a matrix M is a matrix H such that $M = UH$, where U is a unimodular matrix (formed by unimodular column operations) and H is lower triangular, non-negative and each row has a unique maximum entry which is on the main diagonal. Such a matrix H always exists and its columns form a basis of the lattice spanned by the columns of M, because they differ up to unimodular (lattice-preserving) operations. There are many texts on the subject; we refer the reader to the lecture notes of Shmonin [25] for more detailed explanations.

The non-zero columns of a matrix in Hermite normal form constitute a basis of the lattice generated by the columns of the original matrix. Hence a basis of the lattice spanned by a collection of vectors can be obtained by computing the Hermite normal form of the matrix formed by placing the vectors as columns. The Hermite normal form can be computed in polynomial time [26], which we now use to prove Proposition 12.

Proof of Proposition 12 It is equivalent to ask whether $x - x^{(0)} \in \left(p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\right)$. Recall that we can place the lattice into Hermite normal form in polynomial time. That is, determine $d' \leq d, p_1, \ldots, p_{d'}$ such that $p'_1\mathbb{Z} + \cdots + p'_{d'}\mathbb{Z} = p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}$.

As the lattice is in Hermite normal form, there exists a unique choice of $\alpha_1, \ldots, \alpha_{d'}$ such that $\sum_{i=1}^{d'} \alpha_i p_i = x - x^{(0)}$, which can be determined by Gaussian elimination. Then we have $x \in L_i$ if and only if the choices of $\alpha_1, \ldots, \alpha_{d'}$ are integer. □

We now prove the main theorem of this section:

Proof of Theorem 10 We claim that Algorithm 1 returns the strongest $\mathbb{Z}$-linear invariant I in polynomial time. Let us first explain the idea of the algorithm, which proceeds in two phases:

[9] The last replacement is valid, since if $x = y + \beta p_i \in L$ then $x = y + \beta(p_i + \alpha p_j) - \beta\alpha p_j$ is in the new lattice.

- First compute a subset $L_0 \subseteq I$ of the invariant that has the same dimension as I.

  Recall the set $R_0 = \{x^{(0)}, r_1, \ldots, r_{d'}\} \subseteq \mathcal{O}$, with $d' \leq d$, from Lemma 6. The resulting $\mathbb{Z}$-linear set $L_0 = \left(x^{(0)} + (r_1 - x^{(0)})\mathbb{Z} + \cdots + (r_{d'} - x^{(0)})\mathbb{Z}\right)$ is then a d'-dimensional porous subset of the d'-dimensional affine hull of the orbit ($L_0 \subseteq \mathrm{Aff}(\mathcal{O})$). Applying $M_1, \ldots, M_k$ can only increase the density, but not the dimension. As each r_i and $x^{(0)}$ are in $\mathcal{O}$, by Proposition 11 we can assume that each of the directions $(r_i - x^{(0)})$ must be represented in any $\mathbb{Z}$-linear set containing $\mathcal{O}$, and we therefore have that $L_0 \subseteq I$.
- In the second phase, we 'fill in' the lattice as required to cover the whole of $\mathcal{O}$. We compute a growing sequence $L_0 \subsetneq L_1 \subsetneq \cdots \subsetneq L_{m-1} = L_m = I$, where at each step the algorithm merely increases the density of the attendant sets in order to 'fill in' missing points of the invariant.

  To do this we repeatedly find new points which are not yet covered by L_i. Supposing we find $x' \in \mathcal{O} \setminus I$, we then use Proposition 11 to argue that we can add the vector $x' - x^{(0)}$.

□

Claim 14 (Termination) *Algorithm* 1 *terminates*.

Proof of claim The vectors $p_1 = (r_1 - x^{(0)}), \ldots, p_{d'} = (r_{d'} - x^{(0)})$ form a parallelepiped (hyper-parallelogram) that repeats regularly. There are a finite number of integral points inside this parallelepiped. If new points are added in some step, they are added to every parallelepiped. Thus we can add new points finitely many times before saturating or L_i becomes fixed. □

Claim 15 (I is an inductive invariant) *Let* $M \in \{M_1, \ldots, M_k\}$ *and let* $x \in I$. *Then* $Mx \in I$.

Proof of claim It is clear that $x^{(0)} \in I$ as $x^{(0)} \in R_0$.

Let $R = \{r_0, \ldots, r_m\}$ be as in the last iteration of the algorithm, with $r_0 = x^{(0)} \in R$, and so $I = \left(r_0 + \sum_{i=1}^{m}(r_i - r_0)\mathbb{Z}\right)$.

Given a vector $y \in \mathbb{Z}^d$, we denote by $\begin{pmatrix} y \\ 1 \end{pmatrix}$ the vector in $\mathbb{Z}^{d+1}$ formed by y in the first d dimensions and 1 in the final dimension. We first show that for any $y \in \mathbb{Z}^d$:

$$y \in I \iff \begin{pmatrix} y \\ 1 \end{pmatrix} \in \sum_{r \in R} \begin{pmatrix} r \\ 1 \end{pmatrix} \mathbb{Z}. \tag{1}$$

Let $y = \left(r_0 + \sum_{i=1}^{m}(r_i - r_0)a_i\right) \in I$, then $y = \left(r_0(1 - \sum_{r_i} a_i) + \sum_{i=1}^{m} r_i a_i\right)$. Then we have $\begin{pmatrix} y \\ 1 \end{pmatrix} = \begin{pmatrix} r_0 \\ 1 \end{pmatrix}(1 - \sum_{r_i} a_i) + \sum_{i=1}^{m} \begin{pmatrix} r_i \\ 1 \end{pmatrix} a_i \in \sum_{r \in R} \begin{pmatrix} r \\ 1 \end{pmatrix} \mathbb{Z}$. Conversely, let $\begin{pmatrix} y \\ 1 \end{pmatrix} \in \sum_{i=0}^{m} \begin{pmatrix} r_i \\ 1 \end{pmatrix} a_i$, since $\sum_{i=0}^{m} a_i = 1$ then $a_0 = 1 - \sum_{i=1}^{m} a_i$ and we have $\begin{pmatrix} y \\ 1 \end{pmatrix} = \begin{pmatrix} r_0 \\ 1 \end{pmatrix} + \sum_{i=1}^{m} \left(\begin{pmatrix} r_i \\ 1 \end{pmatrix} - \begin{pmatrix} r_0 \\ 1 \end{pmatrix} \right) a_i$, thus in particular, $y = r_0 + \sum_{i=1}^{m}(r_i - r_0)a_i \in I$.

By termination of the algorithm we have $Mr_i \in I$ for all $r_i \in R$ (otherwise Algorithm 1 would add Mr_i to R) and thus $\begin{pmatrix} Mr_i \\ 1 \end{pmatrix} \in \sum_{r_j \in R} \begin{pmatrix} r_j \\ 1 \end{pmatrix} \mathbb{Z}$ for all $r_i \in R$. Let $a_{0,i}, \ldots, a_{n,i} \in \mathbb{Z}$ be such that $\begin{pmatrix} Mr_i \\ 1 \end{pmatrix} = \sum_{r_j \in R} \begin{pmatrix} r_j \\ 1 \end{pmatrix} a_{j,i}$.

By $x \in I$ and Eq. (1) we have $\begin{pmatrix} x \\ 1 \end{pmatrix} = \sum_{r_i \in R} \begin{pmatrix} r_i \\ 1 \end{pmatrix} b_i$ for some $b_0, \ldots, b_n \in \mathbb{Z}$.

Let us now establish that $Mx \in I$. We have $\begin{pmatrix} Mx \\ 1 \end{pmatrix} = \sum_{r_i \in R} \begin{pmatrix} Mr_i \\ 1 \end{pmatrix} b_i$. Therefore we have $\begin{pmatrix} Mx \\ 1 \end{pmatrix} = \sum_{r_i \in R} \sum_{r_j \in R} \begin{pmatrix} r_j \\ 1 \end{pmatrix} a_{j,i} b_i$. Thus $\begin{pmatrix} Mx \\ 1 \end{pmatrix} \in \sum_{r_i \in R} \begin{pmatrix} r_i \\ 1 \end{pmatrix} \mathbb{Z}$, entailing $Mx \in I$ (again by Eq. 1). □

Claim 16 (I is the strongest invariant) *For every invariant J, we have $I \subseteq J$.*

Proof of claim By induction, let us prove that every invariant J must contain L_i. Clearly this is the case for L_0 because all points of $R_0 \subseteq \mathcal{O}$ must be in J and every period vector in L_0 can be present, without loss of generality, thanks to Proposition 11. Assume $L_i \subseteq J$. Then it must be the case that J contains every $M_j(x)$ for $x \in L_i$, as otherwise it would not be an invariant. It therefore follows that J must contain L_{i+1}, since the latter is the minimal $\mathbb{Z}$-linear set containing L_i and $M_j(x)$ for some $j \leq k$. Finally, since I is itself one of the L_i's, we have $I \subseteq J$ as required. □

Claim 17 (Polynomial time) *The algorithm runs in polynomial time in d, k and $\log(\mu)$.*

Proof of claim Let $x \in \mathbb{Z}^d$. We denote by $\|x\|_\infty$ the largest absolute value of an entry of x, and by $\|x\|_2$ the Euclidean norm of the vector.

Recall the parallelepiped from the claim of termination. The volume of the parallelepiped is bounded above by $\|p_1\|_2 \cdots \|p_{d'}\|_2$. The volume of the parallelepiped must at least halve at every step in which a vector is added to the invariant; a new vector either leaves the parallelepiped unchanged, or partitions it into at least two pieces, in which case, one of the two pieces has volume at most half of the original. The volume at step t is therefore $vol_t \leq \|p_1\|_2 \cdots \|p_{d'}\|_2 / 2^t$. The procedure must saturate at, or before, the volume becomes 1, which occurs after at most $\log(\|p_1\|_2 \cdots \|p_{d'}\|_2)$ steps.

Using Lemma 6 we obtain that each $r_i \in R_0$ is the result of at most d matrix multiplication operations; thus using Proposition 7 we have $\|r_i\|_\infty \leq d^d \mu^{d+1}$. Using the triangle inequality, we have $p_i = r_i - x^{(0)}$ we have $\|p_i\|_\infty \leq d^d \mu^{d+1} + \mu \leq (d\mu)^{d+1}$ (for $d \geq 2$).

Using $\|p_i\|_2 \leq \sqrt{d \|p_i\|_\infty}$, we obtain $\|p_i\|_2 \leq \sqrt{d}(d\mu)^{d+1}$. Taking liberal simplifications we obtain $\|p_i\|_2 \leq (d\mu)^{2d}$. Hence $\|p_1\|_2 \cdots \|p_{d'}\|_2 \leq ((d\mu)^{2d})^d$. Hence the number of update steps where a vector is added is at most $\log((d\mu)^{2d^2}) = (2d^2)\log(d\mu)$.

Since the number of vectors is at most $(2d^2)\log(d\mu)$, the number of steps between adding a vector is at most $k(2d^2)\log(d\mu)$ (a new vector is added at least once across all iterations in the inner for loops, otherwise the procedure terminates). Hence, the total number of steps (counting a matrix multiplication, and verifying $x \in L_i$ as a single step) is at most $O(k((2d^2)\log(d\mu))^2)$.

It remains to verify that the bit size of the vectors is polynomial. This will imply that the running time of the matrix multiplications is polynomial as well.

Let (R_i) be the increasing sequence of sets built in Algorithm 1. As there are at most $(2d^2)\log(d\mu)$ vectors added in those sets and at least one vector is added at each step, this sequence becomes stationary after at most $(2d^2)\log(d\mu)$ steps. Given $i \leq (2d^2)\log(d\mu)$, we have that each vector $x' \in R_i$ is the result of $M_\ell x$ for some $x \in R_{i-1}$ and $\ell \in \{1, \ldots, k\}$. Hence, each element $x \in R_i$ is the result of at most i matrix

multiplications. By Proposition 7, after v matrix applications, the size of the number is at most $d^v \mu^{v+1}$. Hence $\|x\|_\infty \leq \mu(d\mu)^{(2d^2)\log(d\mu)}$, thus the bit size of such numbers are at most $(2d^2)\log^2(d\mu) + \log(\mu)$, which is polynomial in d and $\log(\mu)$. □

Claims 14 to 17 conclude that Algorithm 1 computes the strongest inductive invariant I, terminating in polynomial time, as required. □

Remark 18 Considering again the MU puzzle, we note that both 1 and 2 are in the reachability set, hence $(1 + 1\mathbb{Z}) = \mathbb{Z}$ is the strongest $\mathbb{Z}$-linear invariant. Thus the class of $\mathbb{Z}$-linear sets is not useful for certifying non-reachability in this case.

4.2 Extensions of $\mathbb{Z}$-linear sets without strongest invariants

In this section we show that several generalisations of the class of $\mathbb{Z}$-linear sets fail to admit strongest invariants.

$\mathbb{Z}$-semi-linear sets are unions of $\mathbb{Z}$-linear sets, and therefore all finite sets are $\mathbb{Z}$-semi-linear. Consider the deterministic dynamical system starting from point 1 and doubling at each step $\mathcal{M} = (1, (x \mapsto 2x))$. This system has reachability set $\mathcal{O} = \left\{2^k \mid k \in \mathbb{N}\right\}$. For this LDS we can construct the invariant $\left\{2, 4, 8, ..., 2^k\right\} \cup \left\{2^{k+1} p_1 \mid p_1 \in \mathbb{Z}\right\}$ for each k. For any proposed strongest $\mathbb{Z}$-semi-linear invariant, one can find a k for which the corresponding invariant is strictly smaller.

$\mathbb{N}$-linear sets generalise $\mathbb{Z}$-linear sets (observe that $\mathbb{Z}$-linear sets are a proper subclass, since $\left(x + p_i\mathbb{Z}\right)$ can be expressed as $\left(x + (-p_i)\mathbb{N} + p_i\mathbb{N}\right)$, but $\left(x + p_i\mathbb{N}\right)$ is clearly not $\mathbb{Z}$-linear). Consider the LDS $\left((x_1, x_2), \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right)$, with a reachability set consisting of just two points $x = (x_1, x_2)$ and $y = (x_2, x_1)$. There are two incomparable candidates for the minimal $\mathbb{N}$-linear invariant: $(x + (y - x)\mathbb{N})$ and $(y + (x - y)\mathbb{N})$. Similarly for $\mathbb{R}_+$-linear invariants, the sets $\left(y + (x - y)\mathbb{R}_+\right)$ and $\left(x + (y - x)\mathbb{R}_+\right)$ are incomparable half-lines.

5 $\mathbb{N}$-semi-linear invariants

We turn now to $\mathbb{N}$-semi-linear invariants, the most general class of invariants that we consider. $\mathbb{N}$-semi-linear invariants gain expressivity thanks to the 'directions' provided by the period vectors. For example, the only possible $\mathbb{Z}$-semi-linear invariant for the LDS $(0, (x \mapsto x + 1))$ is $\mathbb{Z}$, yet the reachability set, namely $\mathbb{N}$, is $\mathbb{N}$-linear.

In Sect. 5.1 we show that a separating $\mathbb{N}$-semi-linear inductive invariant can *always* be found for unreachable instances of deterministic integer LDS, although the computed invariant will depend on the target (strongest invariants do not always exist here). However, in Sect. 5.2 we show that finding invariants is undecidable for non-deterministic systems, at least in high dimension. Nevertheless, we show in Sect. 5.3 decidability for the low-dimensional setting of the MU Puzzle—one dimension with affine updates.

5.1 Existence of sufficient (but non-minimal) $\mathbb{N}$-semi-linear invariants for point reachability in deterministic LDS

Kannan and Lipton showed decidability of reachability of a point target for deterministic LDS [6]. In this subsection, we establish the following result to provide a separating invariant in unreachable instances.

Theorem 19 *Given a deterministic LDS $(x^{(0)}, M)$ together with a point target y, if the target is unreachable then a separating $\mathbb{N}$-semi-linear inductive invariant can be provided effectively.*

To do so, we will invoke the results from [5] to compute an $\mathbb{R}_+$-semi-linear inductive invariant, and then extract from it an $\mathbb{N}$-semi-linear inductive invariant. More precisely, the authors of [5] show how to build polytopic inductive invariants for certain deterministic LDS. Such polytopes are either bounded or are $\mathbb{R}_+$-semi-linear sets. In the first case, the polytope contains only finitely many integral points, which can directly be represented via an $\mathbb{N}$-semi-linear set. In the second case, we build an $\mathbb{N}$-semi-linear set containing exactly the set of integral points included in the $\mathbb{R}_+$-semi-linear invariant, thanks to the following lemma.

Lemma 20 *Given an $\mathbb{R}_+$-linear set $S = \left(x + \sum_i p_i \mathbb{R}_+\right)$, where the vectors p_i have rational coefficients and x is an integer vector, one can build an $\mathbb{N}$-semi-linear set N comprising precisely all of the integral points of S.*

Proof Let $S = \left(x + \sum_i p_i \mathbb{R}_+\right)$ be a $\mathbb{R}_+$-linear set where the vectors p_i have rational coefficients and x is an integer vector. Let $k \in \mathbb{N}$ be an integer so that the vectors kp_i have integer coefficients. We denote by v_j the integer vectors of the form $\sum_i \mu_i k p_i$ where $0 \leq \mu_i \leq 1$. Then the set $T = \left(x + \sum_j v_j \mathbb{N}\right)$ contains exactly the integer vectors contained in S.

Indeed, first T only contains integer vectors since both x and the vectors v_j are integer vectors. Secondly, all the vectors in T are included in S as the period vectors of T lie in the cone defined by the vectors of S. Finally, given an integer vector y in S, y can be rewritten as $y = x + v + \sum_i m_i k p_i$ where for all $i, m_i \in \mathbb{N}$ and v is of the form $\sum_i \mu_i k p_i$ with $0 \leq \mu_i \leq 1$ Therefore there exists j such that $v_j = v$ and as for all i, kp_i is a period vector of T, $y \in T$. □

Proof of Theorem 19 We note that every inductive invariant produced in [5] has rational period vectors, as the vectors are given by the difference of successive points in the orbit of the system, and thus Lemma 20 can be applied. This produces an inductive invariant as their invariant is inductive, the LDS only reaches integer vectors and the invariant produced through Lemma 20 contains all the integer points appearing within their invariant.

The authors of [5] build an inductive invariant in all cases except those for which every eigenvalue of the matrix governing the evolution of the LDS is either 0 or of absolute value 1 and at least one of the latter is not a root of unity. This situation however cannot occur in our setting. Indeed, the eigenvalues of an integer matrix are algebraic integers, and an old result of Kronecker [27] asserts that unless all of the eigenvalues are roots of unity, one of them must have absolute value strictly greater than 1 (the case in which *all* eigenvalues are 0 being of course trivial). □

5.2 Undecidability of $\mathbb{N}$-semi-linear invariants for nondeterministic LDS

If the enhanced expressivity of $\mathbb{N}$-semi-linear sets allows us to always find an invariant for deterministic LDS, it contributes in turn to making the invariant-synthesis problem undecidable when the LDS is not deterministic.

We establish this through a reduction from the infinite Post correspondence problem (ω-PCP), which can be defined in the following way: given m pairs of non-empty words $\{(u^1, v^1), \dots, (u^m, v^m)\}$ over a binary alphabet, does there exist an infinite word $w = w_1 w_2 \dots$ over alphabet $\{1, \dots, m\}$ such that $u^{w_1} u^{w_2} \dots = v^{w_1} v^{w_2} \dots$. This problem is known to be undecidable when m is at least 8 [28, 29].

Theorem 21 *The invariant synthesis problem for $\mathbb{N}$-semi-linear sets and linear dynamical systems with* 13 *matrices of dimension* 7, *or two matrices of dimension* 91, *is undecidable.*

Proof This proof follows in part the structure of the argument showing the undecidability of the invariant synthesis problem for $\mathbb{R}_+$-semi-linear invariants presented in [5]. Some non-trivial changes and new ideas have to be added here due to the restriction to integer values.

We will transform an instance of ω-PCP with m tiles to an instance of the invariant synthesis problem for $m + 5$ matrices of size 7. This can then be converted in routine fashion to an instance of two matrices of size $7m + 35$ (see Theorem 9 of [5] for instance).

The main idea of this proof is to encode a pair of words on alphabet $\{1, 2\}$ corresponding to each sequence of tiles as an integer in base 4. An important property of our encoding is that the operation of appending a new tile to an existing pair of words can be achieved by matrix multiplication.

Recall that if the instance of ω-PCP is negative, then every generated pair of words will differ at some point. Our reduction is such that a difference of letters creates a difference in their numerical encodings that can be identified through an $\mathbb{N}$-semi-linear invariant. Conversely, when the ω-PCP instance has a positive answer, there can be no $\mathbb{N}$-semi-linear invariant.

Short simplifying lemma

In order to simplify the main part of the proof, let us first show that one can enforce an order between the matrices using affine transformations on one dimension. Let us denote by p this dimension; it is initially equal to 1 and its target value is 0. Consider the three following affine transformations: $f_1(p) = 2p - 1$, $f_2(p) = 2p - 2$ and $f_3(p) = 2p$. The only sequences of transformations allowing to reach the target are of the form $f_3^* f_2 f_1^*$. Indeed, let $\mathcal{I} = \{p \mid p \geq 2 \vee p \leq -1\}$, we have (1) if $p \in \mathcal{I}$, then for all $i \in \{1, 2, 3\}$, $f_i(p) \in \mathcal{I}$, (2) $f_1(1) = 1$ and $f_1(0) \in \mathcal{I}$, (3) $f_2(1) = 0$ and $f_2(0) \in \mathcal{I}$ and (4) $f_3(1) \in \mathcal{I}$ and $f_3(0) = 0$. As a consequence, the inductive invariant $\mathcal{I}$ ensures that any sequence of transformations that do not have the desired order cannot reach the target. In the following, we will call type 1, 2 or 3 the transformations we define, depending on whether they implicitly contain the function f_1, f_2 or f_3.

Description of the reduction

We reduce an instance $\{(u^1, v^1), \dots, (u^m, v^m)\}$ of the ω-PCP problem over binary alphabet $\{1, 2\}$ to the invariant synthesis problem. Given a finite or infinite word w, we denote by $|w|$ the length of the word w and given an integer $i \leq |w|$, we write w_i for the *i-th* letter of w. Given a finite or infinite word w on alphabet $\{1, \dots, m\}$ we denote by u^w and v^w the words on alphabet $\{1, 2\}$ such that $u^w = u^{w_1} u^{w_2} \dots$ and $v^w = v^{w_1} v^{w_2} \dots$. Given a finite word w on alphabet $\{1, 2\}$, denote by $[w] = \sum_{i=1}^{|w|} w_i 4^{|w|-i}$ the quaternary encoding of w. It is clear that it satisfies $[ww'] = 4^{|w'|}[w] + [w']$. For all $i \leq m$, we denote by $n_i = 4^{|u^i|}$, $m_i = 4^{|v^i|}$ and $\mathsf{max}_i = \max(n_i, m_i)$.

We work with five dimensions, (s, c, d, n, k), and define the following transformations:

- For $i \leq m$, the type 1 transformation $\mathsf{Simulate}_i$ on (s, c, d, n, k) encodes the action of reading the pair (u^i, v^i) and increases the counters n and k: it simultaneously applies $s \leftarrow \mathsf{max}_i s + c[u^i]\frac{\mathsf{max}_i}{n_i} - d[v^i]\frac{\mathsf{max}_i}{m_i}$, $c \leftarrow \frac{\mathsf{max}_i}{n_i} c$, $d \leftarrow \frac{\mathsf{max}_i}{m_i} d$, $n \leftarrow n + k$ and $k \leftarrow k + 1$.
- The type 2 transformation $\mathsf{Transfer}$ on (s, c, d, n, k) gathers some of the values in order to compare them and resets d: $s \leftarrow s - c - d$, $c \leftarrow -s - c - d$ and $d \leftarrow 0$.
- The type 3 transformation $\mathsf{Inc_s}$ increments s: $s \leftarrow s + 1$.
- The type 3 transformation $\mathsf{Inc_c}$ increments c: $c \leftarrow c + 1$.
- The type 3 transformation Dec decreases k and n: $n \leftarrow n - k$, $k \leftarrow k - 1$.
- The type 3 transformation $\mathsf{Dec_k}$ decrements k: $k \leftarrow k - 1$.

These $m + 5$ transformations operate over seven dimensions in total: the five above (namely (s, c, d, n, k)), one (namely p) for ordering the transformations, and one last dimension constantly equal to 1, required to implement affine transformations.

We will show that there is a solution to the given instance of the ω-PCP problem iff there does not exist an $\mathbb{N}$-semi-linear invariant for the system with initial point $x = (0, 1, 1, 0, 0, 1, 1)$, target $y = (0, 0, 0, 1, 0, 0, 1)$, and using the matrices inducing the transformations defined above.

Evolution of the system

Let $w = w_1 \dots w_j$ be a finite word over $\{1, \dots, m\}^*$. Consider $(s, c, d, n, k, p, a) = \mathsf{Simulate}_w x$ where $\mathsf{Simulate}_w$ represents the transformation $\mathsf{Simulate}_{w_j} \dots \mathsf{Simulate}_{w_2} \mathsf{Simulate}_{w_1}$. We have

- $s = c[u^w] - d[v^w]$,
- $n = \frac{j(j-1)}{2}$ and $k = j$,
- $p = a = 1$.

Indeed, let us prove the first item (the only non-trivial one) by induction on the length of w. If $|w| = 0$, then $[u^w] = [v^w] = 0$ which is compatible as the first component of x is 0. Otherwise, w is of the form zi with $i \in \{1, \dots, m\}$. By the induction hypothesis, denoting $(s, c, d, n, k, p, a) = \mathsf{Simulate}_w x$ and $(s', c', d', n', k', p', a') = \mathsf{Simulate}_z x$, we have that $s' = c'[u^z] - d'[v^z]$. Applying $\mathsf{Simulate}_i$, we obtain that $s = \mathsf{max}_i s' + c'[u^i]\frac{\mathsf{max}_i}{n_i} - d'[v^i]\frac{\mathsf{max}_i}{n_i}$, $c = \frac{\mathsf{max}_i}{n_i} c'$ and $d = \frac{\mathsf{max}_i}{m_i} d'$. Thus

$$\begin{aligned}
s =& \mathsf{max}_i(c'[u^z] - d'[v^z]) + c'[u^i]\frac{\mathsf{max}_i}{n_i} - d'[v^i]\frac{\mathsf{max}_i}{m_i} \\
=& c'(\mathsf{max}_i[u^z] + [u^i]\frac{\mathsf{max}_i}{n_i}) - d'(\mathsf{max}_i[v^z] + [v^i]\frac{\mathsf{max}_i}{m_i}) \\
=& c(n_i[u^z] + [u^i]) - d(m_i[v^z] + [v^i]) \\
=& c[u^w] - d[v^w]
\end{aligned}$$

which concludes the induction.

Only if case: ω-PCP solution implies no invariant

Assume that there is a solution w to the ω-PCP instance. Consider the sequence of points (x_n) obtained as follows: for all $j \in \mathbb{N}$, denoting $w_{\leq j}$ the prefix of w of length j, $x_j = (s_j, c_j, 0, n_j, k_j, 0, 1) = \mathsf{Transfer\ Simulate}_{w_{\leq j}} x$.

Let (s, c, d) be the three first components of $\mathsf{Simulate}_{w_{\leq j}} x$. Assuming without loss of generality that $|u^{w_{\leq j}}| \leq |v^{w_{\leq j}}|$ we have that

$$\begin{aligned}
|s| &= |c[u^{w_{\leq j}}] - d[v^{w_{\leq j}}]| \\
&= \sum_{i=1}^{|u^{w_{\leq j}}|} |u_i^{w_{\leq j}} - v_i^{w_{\leq j}}| c4^{|u^{w_{\leq j}}|-i} + \sum_{i=|u^{w_{\leq j}}|+1}^{|v^{w_{\leq j}}|} v_i^{w_{\leq j}} c4^{|u^{w_{\leq j}}|-i} \\
&= \sum_{i=|u^{w_{\leq j}}|+1}^{|v^{w_{\leq j}}|} v_i^{w_{\leq j}} c4^{|u^{w_{\leq j}}|-i} \\
&< c\,.
\end{aligned}$$

The first equality was proven in the previous paragraph. The second equality is obtained by grouping the terms corresponding to the same power of 4 and noting that, by construction, $c4^{|u^{w_{\leq j}}|} = d4^{|v^{w_{\leq j}}|}$. The third equality comes from the fact that $w_{\leq j}$ is a prefix of a solution to the ω-PCP instance and thus that letters on the same level are the same. Finally, the last inequality is obtained by bounding every $v_i^{w_{\leq j}}$ by 2 and extending the sum to infinity.

From this inequality, we immediately have that $|s| - c - d$ is negative, and thus both $s_j = s - c - d$ and $c_j = -s - c - d$ are negative.

Due to the above, by applying to the points x_j a number of times the transformations $\mathsf{Inc_s}$ and $\mathsf{Inc_c}$, we obtain the sequence of points $(y_j)_{j\in\mathbb{N}}$ where $y_j = (0, 0, 0, n_j, k_j, 0, 1)$. We claim that any semi-linear invariant containing all the points y_j also contains a point of the form $(0, 0, 0, n_j + d, k_j, 0, 1)$, where d is a positive integer. This will imply the result as from such a point, one can reach the target by $d - 1$ applications of $\mathsf{Dec_k}$ and k_j applications of Dec and thus there is no semi-linear invariant of the system that does not intersect the target.

Let us now prove the above claim. Let $\mathcal{I}$ be a semi-linear set containing every vector y_j (which we will see as two-dimensional objects by projecting on the 4th and 5th dimension). Then there exists a linear set $\mathcal{I}' \subseteq \mathcal{I}$ that contains infinitely many vectors of $(y_j)_{j\in\mathbb{N}}$. This set $\mathcal{I}'$ is defined by an initial vector, and a set of period vectors. As $\mathcal{I}'$ contains infinitely many vectors of $(y_j)_{j\in\mathbb{N}}$ where the ratios between the first and second component is increasing, one of the period vectors is of the form $(d, 0)$ where d is a strictly positive integer. Let j be such that $y_j \in \mathcal{I}'$, then $(n_j + d, k_j) \in \mathcal{I}'$ which implies the claim.

As a consequence, every inductive $\mathbb{N}$-semi-linear invariant of the LDS intersects with the target.

If case: no ω-PCP solution implies an invariant

Assume that there is no solution to the ω-PCP instance. There exists $n_0 \in \mathbb{N}$ such that for every infinite word w on alphabet $\{0, \dots, m\}$ there exists $n \leq n_0$ such that $u_n^w \neq v_n^w$. Indeed, consider the tree whose root is labelled by $(\varepsilon, \varepsilon)$ and, given a node (u, v) of the tree, if for all $n \leq \min(|u|, |v|)$ we have $u_n = v_n$, then this node has m children: the nodes (uu^i, vv^i) for $i \in \{1, \dots, m\}$. This tree is finitely branching and does not contain any infinite path (which would induce a solution to the ω-PCP instance). Thus, according to König's lemma, it is finite. We can therefore choose the height of this tree as our n_0.

We define the invariant $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$ where

$$\begin{aligned}\mathcal{I}_1 =& \left\{\mathsf{Simulate}_w(x) \mid w \in \{1, \dots, m\}^* \wedge |w| \leq n_0 + 1\right\},\\ \mathcal{I}_2 =& \left\{z = (s, c, 0, n, k, 0, 1) \mid z = (\mathsf{Inc_s})^*(\mathsf{Inc_c})^*(\mathsf{Dec})^*(\mathsf{Dec_k})^*\mathsf{Transfer}\,\mathsf{Simulate}_w(x)\right.\\ &\left. \wedge\, w \in \{1, \dots, m\}^* \wedge |w| \leq n_0 + 1 \wedge s, t, n, k \in \mathbb{N}\right\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{I}_3 =& \left\{(s, c, d, n, k, p, 1) \mid (|s| - c - d \geq 1 \wedge c \geq 0 \wedge d \geq 0 \wedge p = 1)\right.\\ &\left. \vee\, ((s \geq 1 \vee c \geq 1 \vee n \leq -1 \vee k \leq -1) \wedge p = 0) \vee p \leq -1 \vee p \geq 2\}.\right\}\end{aligned}$$

By definition, $\mathcal{I}$ is an $\mathbb{N}$-semi-linear set, contains x and does not contain y. The difficulty is to show stability under the transformations.

$\diamond$ Let $z = \mathsf{Simulate}_w(x) \in \mathcal{I}_1$, for some $w \in \{1, \dots, m\}^*$ with $|w| \leq n_0 + 1$. By ordering if we apply a transformation outside $\mathsf{Transfer}$ or a $\mathsf{Simulate}_i$ for some i, we reach $\mathcal{I}_3$.

- For $i \in \{1, \dots, m\}$, if $|w| \leq n_0$, then $\mathsf{Simulate}_i z \in \mathcal{I}_1$.

Else, $\mathsf{Simulate}_i z = \mathsf{Simulate}_{wi} x = (s, c, d, n, k, p, 1)$ with $|w| = n_0 + 1$. But then, there exists $n_1 \leqslant n_0$ such that $u_{n_1}^{wi} \neq v_{n_1}^{wi}$. Let n_2 be the smallest such number, then assume without loss of generality that $c \geq d$, we have

$$\begin{aligned}s &= c[u^{wi}] - d[v^{wi}]\\ &= (u_{n_2}^{wi} - v_{n_2}^{wi})c4^{|u^{wi}|-n_2} + \sum_{j=n_2+1}^{\max(|u^{wi}|,|v^{wi}|)} (u_j^{wi} - v_j^{wi})c4^{|u^w i|-j}\end{aligned}$$

since $u_j^{wi} = v_j^{wi}$ for $j < n_2$. Thus,

$$\begin{aligned}|s| &\geqslant c4^{|u^{wi}|-n_2} - \frac{2c}{3}4^{|u^{wi}|-n_2} && \text{since } |u_{n_2}^{wi} - v_{n_2}^{wi}| = 1\\ & && \text{and for } n \geq n_2, |u_n^{wi} - v_n^{wi}| \leq 2\\ &\geqslant \frac{1}{3}c4^{|u^{wi}|-n_2}\\ &\geqslant 2c + 1 && \text{since } n_2 \leqslant n_0 \text{ and } |u^{wi}| \geqslant n_0 + 2.\end{aligned}$$

As $c \geq d$, this shows that $\mathsf{Simulate}_i z \in \mathcal{I}_3$.

- $\mathsf{Transfer} z \in \mathcal{I}_2$.

$\diamond$ Let $z \in \mathcal{I}_2$ and f be one of the transformations, then $f(z) \in \mathcal{I}_2$ if f increased (resp. decreased) a negative (resp. positive) component. Otherwise $f(z) \in \mathcal{I}_3$.

⋄ Let $z = (s, c, d, n, k, p, 1) \in \mathcal{I}_3$, f be one of the transformations and $f(z) = (s', c', d', n', k', p', 1)$.

- if $p = 0$, then either $p' \leq -1$ and $f(z) \in \mathcal{I}_3$ or z satisfies $(s \geq 1 \vee c \geq 1 \vee n \leq -1 \vee k \leq -1)$ and then $f(z)$ satisfies $(s' \geq 1 \vee c' \geq 1 \vee n' \leq -1 \vee k' \leq -1)$, thus $f(z) \in \mathcal{I}_3$.
- if $p = 1$, then $|s| - c - d \geq 1, c \geq 0$ and $d \geq 0$. There are three possibilities (1) $p' = 2$ and thus $f(z) \in \mathcal{I}_3$, (2) $f = \mathsf{Transfer}$ then $p' = 0$ and either $s' \geq 1$ or $c' \geq 1$ and thus $f(z) \in \mathcal{I}_3$ or (3) $f = \mathsf{Simulate}_i$ for $i \leq m$. In the latter case without loss of generality, assume that $d' \geqslant c'$. We have that

$$
\begin{aligned}
\left|s'\right| &= |\mathsf{max}_i s + c'[u^i] - d'[v^i]| && \text{by applying } \mathsf{Simulate}_i \\
&\geqslant \mathsf{max}_i |s| - d' \max([u^i], [v^i]) \\
&\geqslant \mathsf{max}_i (c + d + 1) - d' \max([u^i], [v^i]) && \text{by assumption on } |s| \\
&\geqslant \mathsf{max}_i (c + d + 1) - \tfrac{2}{3} d \mathsf{max}_i && \text{since } [u^i] \in [0, \tfrac{2n_i}{3}] \\
&= \mathsf{max}_i (c + d/3) + \mathsf{max}_i \\
&\geqslant c' + d' + 1
\end{aligned}
$$

since $\mathsf{max}_i c \geq c'$, $\mathsf{max}_i d/3 \geq d'$ (as $m_i \geq 4$) and $\mathsf{max}_i \geqslant 4$. This shows that $f(z) \in \mathcal{I}_3$. Therefore $\mathcal{I}$ is inductive and thus a $\mathbb{N}$-semi-linear invariant of the system. This concludes the reduction. □

5.3 Nondeterministic one-dimensional affine updates

The previous section shows that point reachability for nondeterministic LDS is undecidable once there are sufficiently many dimensions, motivating an analysis at lower dimensions. The MU Puzzle requires a single dimension with affine updates (or equivalently two dimensions in matrix representation, with the coordinate along the second dimension kept constant). We consider this one-dimensional affine-update case, and therefore, rather than taking matrices as input, we directly work with affine functions of the form $f_i(x) = a_i x + b_i$.

Theorem 22 *Given $x^{(0)}, y \in \mathbb{Z}$, along with a finite set of functions $\{f_1, \ldots, f_k\}$ where $f_i(x) = a_i x + b_i$, $a_i, b_i \in \mathbb{Z}$ for $1 \leq i \leq k$, it is decidable whether y is reachable from $x^{(0)}$. Moreover, when y is unreachable, an $\mathbb{N}$-semi-linear separating inductive invariant can be algorithmically computed in pseudo-polynomial time.*

We note that decidability of reachability is already known [14, 15]. We refine this result by exhibiting an inductive invariant which can be used to certify non-reachability. In fact our procedure will produce an $\mathbb{N}$-semi-linear set which can be used to decide reachability, and which, in instances of non-reachability, will be a separating inductive invariant. We have implemented this algorithm into our tool POROUS, enabling us to efficiently tackle the MU Puzzle as well as its generalisation to arbitrary collections of one-dimensional affine functions. We report on our experiments in Sect. 7.

We build a case distinction depending on the type of functions that appear:

Definition 23 Consider an affine function $f(x) = ax + b$. We say:

- f is *redundant* if $f(x) = b$, (including possibly $b = 0$), or if $f(x) = x$.
- f is a *counter* if $f(x) = x + b$, $b \neq 0$. Two counters $f(x) = x + b$ and $g(x) = x + c$ are *opposing* if $bc < 0$. Otherwise they are called codirectional.
- f is *growing* if $f(x) = ax + b$ and $|a| \geq 2$. We say a growing function is *inverting* if $a \leq -2$.
- f is *pure inverting* if $f(x) = -x + b$ (including possibly $b = 0$).

5.3.1 Simplifying assumptions

Lemma 24 *We can reduce the computation of an invariant for a system having redundant functions to finitely many invariant computations for systems having no such functions.*

Proof Clearly the identity function has no impact on the reachability set, and so can be removed outright. For any other redundant function, its impact on the reachability set does not depend on when the function is used, and we may therefore assume that it was used in the first step, or equivalently, using an alternative starting point. Hence the invariant-computation problem can be reduced to finitely many instances of the problem over different starting points, with redundant functions removed. Finally, taking the union of the resulting invariants yields an invariant for the original system. □

Lemma 25 *Without loss of generality*, $x^{(0)} \geq 0$.

Proof Suppose $x^{(0)} < 0$, we construct a new system, where each transition $f(x) = ax + b$ is replaced by $\bar{f}(x) = ax - b$. Then $x^{(0)}$ reaches y in the original system if and only if $-x^{(0)}$ reaches $-y$ in the new system. To see this, observe that if $f(x) = ax + b$, then $\bar{f}(-x) = -ax - b = -f(x)$. □

Lemma 26 *Suppose there are at least two distinct pure inverting functions* (*and possibly other types of functions*). *Then without loss of generality there are two opposing counters.*

Proof Consider $f(x) = -x + b$, and $g(x) = -x + c$. Then $f(g(x)) = -(-x + c) + b = x + b - c$ and $g(f(x)) = -(-x + b) + c = x + c - b$. Since $b - c = -(c - b)$ and $b \neq c$ (as $f \neq g$) these two functions are opposing. □

5.3.2 Two opposing counters

Let us first observe that when there are two opposing counters, we can essentially move in either direction by some fixed amount. This will entail that only $\mathbb{Z}$-(semi)-linear invariants need be produced, rather than proper $\mathbb{N}$-(semi)-linear invariants.

Lemma 27 *Suppose there are two opposing counters*, $f(x) = x + b$, *and* $g(x) = x - c$. *Then for any reachable x we have* $(x + d\mathbb{Z}) \subseteq I$ *for* $d = \gcd(b, c)$.

Lemma 28 *For ℓ, k coprime, the sequence $a_n = (n\ell \mod k)$ for $n \in \mathbb{N}$ cycles through every residue class $\{0, \ldots, k-1\}$.*

Proof Any path longer than k visits some class twice, and if the shortest cycle is k, then it visits every class.

Suppose there is a cycle of length less than k; then $n\ell = c + mk$ and $(n+i)\ell = c + m'k$ and hence $i\ell = (m'-m)k$, with $i < k$. Since ℓ is an integer i divides $(m'-m)k$ then $i = pr$ for $p, r \in \mathbb{N}$ such that $\frac{m'-m}{p}$ is integer and $\frac{k}{r}$ is integer. Observe that since $r \leq i < k$ we have $\frac{k}{r} > 1$. But this implies that $\frac{k}{r}$ divides k and ℓ, contradicting $\gcd(k, \ell) = 1$. □

Proof of Lemma 27 Let $b = kd, c = \ell d$, where k, ℓ are co-prime.

We show there exists $m, n \geq 0$ such that $mb - cn = d$. We have $mb - cn = d \iff mkd - n\ell d = d \iff mk - n\ell = 1$. Then choose $m = \frac{1+n\ell}{k}$. By Lemma 28 n can be chosen such that $n\ell \equiv k \mod d$ for any $k \in \{0, \ldots, d-1\}$. Then n can be chosen such that $1 + n\ell \equiv 0 \mod d$ and so k divides $1 + n\ell$ for some n.

Hence for $x \in \mathcal{O}$, the set $(x + d\mathbb{N})$ is included in the reachability set: we obtain $x + jd$, $j > 0$ by $g^{nj} \circ f^{mj}(x)$, hence $x + jd \in \mathcal{O}$ and thus $x + d\mathbb{N} \subseteq I$. Similarly, we can find $m', n' \geq 0$ such that $m'b - cn' = -d$ and thus $(x + d\mathbb{Z})$ is also within the reachability set.

□

Therefore, starting with $\left(x^{(0)} + d\mathbb{Z}\right) \subseteq I$ we can 'saturate' the invariant under construction using the following lemma:

Lemma 29 *Let $h(x) = x + d$ be chosen as a reference counter amongst the counters. If $(x + d\mathbb{Z}) \subseteq I$, then $(f(x) + d\mathbb{Z}) \subseteq I$ for every function f.*

Proof of Lemma 29 Consider the function $f(x) = ax + b$. If $x + dk \in I$ for $k \in \mathbb{Z}$, then $f(x + dk) = a(x + dk) + b = ax + adk + b = f(x) + adk \in I$.

Now applying the counter $h(x) = x + d$ an arbitrary number m of times, we have $h^m \circ f(x + dk) = f(x) + adk + dm \in I$ for $k \in \mathbb{Z}$ and $m \in \mathbb{N}$. Thus $f(x) + dn \in I$ for any choice of $n \in \mathbb{Z}$ by suitable choice of k (possibly negative) and m (non-negative). □

Without loss of generality if $(x + d\mathbb{Z})$ is in the invariant, then $0 \leq x < d$. We then repeatedly use Lemma 29 to find the required elements of the invariant. Since there are only finitely many residue classes (modulo d), every reachable residue class $\left(c_1, \ldots, c_n\right)$ can be found by saturation (in at most d steps), yielding invariant $\left(c_1 + d\mathbb{Z}\right) \cup \cdots \cup \left(c_n + d\mathbb{Z}\right)$.

Thanks to Lemma 26, in all remaining cases there is without loss of generality at most one pure inverter.

5.3.3 Only pure inverters

If there is exactly one pure inverter $f(x) = -x + b$ (and no other functions of any type), then $f(x^{(0)}) = -x^{(0)} + b$ and $f(-x^{(0)} + b) = x^{(0)} - b + b = x^{(0)}$, thus the reachability set is $\{x^{(0)}, -x^{(0)} + b\}$, which is itself a finite inductive invariant.

5.3.4 No counters

If we are not in the preceding case and there are no counters, then there must be growing functions and by Lemma 26, without loss of generality at most one pure inverter. We show that all growing functions increase the absolute value outside of some bounded region.

Lemma 30 *For every $M \geq 0$ and every growing function $f(x) = ax + b$, $|a| \geq 2$, there exists $C_f^M \geq 0$ such that if $|x| \geq C_f^M$ then $|f(x)| \geq |x| + M$.*

Proof By the triangle inequality we have: $|f(x)| = |ax + b| \geq |a||x| - |b|$. Thus $|x| \geq \frac{|b|+|M|}{|a|-1} \implies |a||x| - |b| \geq |x| + |M| \implies |f(x)| \geq |x| + M$. □

This is the only situation in which the invariant is not exactly the reachability set, and requires us to take an overapproximation.

Let $C = \max\left\{C_{f_1}^0, \ldots, C_{f_k}^0, |y| + 1\right\}$, for $f_1, \ldots, f_k$ growing functions and y the target point. If there are no pure inverters then $(-C - \mathbb{N}) \cup (C + \mathbb{N})$ is inductive. However, as it may not yet contain $x^{(0)}$, it does not yet contain the whole of $\mathcal{O}$. From this we can build the inductive invariant $(-C - \mathbb{N}) \cup (C + \mathbb{N}) \cup (\mathcal{O} \cap (-C, C))$. The set $\mathcal{O} \cap (-C, C)$ is finite and can be elicited by exhaustive search, noting that once an element of the orbit reaches absolute value at least C, the remainder of the corresponding trajectory remains forever outside of $(-C, C)$.

If there is one pure inverter $g(x) = -x + d$ then observe that $-C$ is mapped to $C + d$ and $C + d$ is mapped to $-C$. Thus intuitively we want to use the interval $(-C, C + d)$. However two problems may occur: (a) since d could be less than 0 then $C + d$ may no longer be growing (under the application of the growing functions), and (b) an inverting growing function only ensures that $-C$ is mapped to a value greater than or equal to C, rather than $C + d$. Hence, we choose C' to ensure that $C' \pm d$ is still growing by at least $|d|$ (under the application of our growing functions). Let $C' = \max\left\{C_{f_1}^{|d|}, \ldots, C_{f_k}^{|d|}, |y| + 1\right\} + |d|$. Then the invariant is $\left(-C' - \mathbb{N}\right) \cup \left(C' + d + \mathbb{N}\right) \cup (\mathcal{O} \cap (-C', C' + d))$.

5.3.5 Codirectional counters

The only remaining possibility (if there do not exist two opposing counters, and not all functions are growing or pure inverters), is that there are counter functions, but they are all codirectional. There may also be a single pure inverter, and any number of growing functions. Throughout this section we assume the growing functions are growing outside of the interval $[-B, C]$.

We pick a counter $h(x) = x + d$ amongst the codirectional counters to be the reference counter; the choice is arbitrary, but it is convenient to pick a counter with minimal $|d|$. For each residue r modulo d, we will have either a set $(r + d\mathbb{Z})$, a set $\left(x_r + d\mathbb{N}\right)$ for $x_r \equiv r \mod d$, or $\emptyset$. We will define a saturation procedure on these sets. To start, clearly we have $\left(x^{(0)} + d\mathbb{N}\right) \subseteq I$.

As in the case of two opposing counters, by Lemma 29, $\mathbb{Z}$-linear sets will induce new $\mathbb{Z}$-linear sets. We now observe that using inverters $\mathbb{N}$-linear sets may induce $\mathbb{Z}$-linear sets:

Lemma 31 *If there is an inverter* $g(x) = -ax + b$, *with* $a > 0, b \in \mathbb{Z}$, *and we have* $(x + d\mathbb{N}) \subseteq I$ *then* $(g(x) + d\mathbb{Z}) \subseteq I$.

Proof Let $r = g(x) + dm$ for $m \in \mathbb{Z}$. We show $r \in I$. Consider $x + dn$ for $n \in \mathbb{N}$, then $g(x + dn) = -a(x + dn) + b = -ax + b - adn = g(x) - adn$. Hence $g(x) - adn + dk$, $n, k \in \mathbb{N}$, is reachable by applying k times the function $h(x)$. Then we have for any $m \in \mathbb{Z}$ there exists $k, n \in \mathbb{N}$ such that $k - na = m$, so that r is indeed reachable. □

Lemma 32 *Let f be a non-inverting function and suppose $h(x) = x + d$ is a counter. If the $\mathbb{N}$-linear set $\{x_r + d\mathbb{N}\}$ is in the invariant, then the set $\{f(x_r) + d\mathbb{N}\}$ is in the invariant.*

There are finitely many $\mathbb{Z}$-linear sets, thus a saturation procedure applied to these sets will terminate. However, repeated application of Lemma 32 will not necessarily saturate. If the application of f to x_r 'moves' in the same direction as the counters then saturation will occur. However, when the function f moves in the opposite direction, we may generate infinitely many such classes. Note that all the counters are assumed to move in same direction as the reference counter (as we do not have opposing counters). However, the direction of a growing function depends on the sign of the input.

Example 33 Consider the reference counter $h(x) = x + 4$, with initial point 5. This yields an initial set $(5 + 4\mathbb{N}) \subseteq \mathcal{O}$, where 5 is the initial point and $4\mathbb{N}$ is derived from the counter increment. Now when applying $x \mapsto 2x + 6$ to $(5 + 4\mathbb{N})$ we obtain $(10 + 6 + 8\mathbb{N} + 4\mathbb{N}) = (16 + 4\mathbb{N})$, then $(38 + 4\mathbb{N})$, and then $(82 + 4\mathbb{N})$. However $(82 + 4\mathbb{N}) \subseteq (38 + 4\mathbb{N})$ and we can therefore stop with the invariant $(5 + 4\mathbb{N}) \cup (16 + 4\mathbb{N}) \cup (38 + 4\mathbb{N})$.

However, if the initial sequence is not moving in the direction of the reference counter, this saturation does not occur. Consider $(5 + 4\mathbb{N})$ with the function $x \mapsto 2x - 6$. Then $(5 + 4\mathbb{N})$ maps to $(10 - 6 + 8\mathbb{N} + 4\mathbb{N}) = (4 + 4\mathbb{N})$, which maps to $(2 + 4\mathbb{N})$, $(-2 + 4\mathbb{N})$, $(-10 + 4\mathbb{N})$, $(-26 + 4\mathbb{N})$, and so on. However -2 and -10 are both 2 modulo 4 (and so is -26 as well). This means in the negative direction we can obtain arbitrarily large negative values congruent to 2 modulo 4 and then use the reference counter $h(x) = x + 4$ to obtain any value of $(2 + 4\mathbb{Z})$.

Finally, we will use the following lemma to induce a $\mathbb{Z}$-linear set when an infinite sequence of $\mathbb{N}$-linear sets occur. Since inverting induces $\mathbb{Z}$-linear sets, in the following lemma we can assume all functions are non-inverting.

Lemma 34 *Assume the reference counter has the form $h(x) = x + d$. Suppose all growing functions are growing outside of $[-B, C]$.*

If $d \geq 0$ and there exist $x_r < -B$ and a sequence of functions $h_1, h_2, \ldots, h_m \in \{f_1, \ldots, f_k\}$ such that

$$h_j \circ \ldots \circ h_1(x_r) < x_r \leq -B \text{ for all } j \leq m \text{ and } h_m \circ \ldots \circ h_1(x_r) \equiv x_r \bmod d,$$

then for all $M \leq x_r$, there exist $h'_1, h'_2, \ldots, h'_{m'}$ such that

$$x_M = h'_{m'} \circ \ldots \circ h'_1(x_r) \leq M \quad \text{and} \quad x_M \equiv x_r \bmod d. \tag{2}$$

Furthermore, if $x_r \in I$, *then* $\left(x_r + d\mathbb{Z}\right) \subseteq I$.

Symmetrically, if $d < 0$ *and there exist* $x_r > C$ *and* $h_1, h_2, \ldots, h_m \in \{f_1, \ldots, f_k\}$ *such that*

$$h_j \circ \ldots \circ h_1(x_r) > x_r \geq C \text{ for all } j \leq m \text{ and } h_m \circ \ldots \circ h_1(x_r) \equiv x_r \bmod d,$$

then for all $M \geq x_r$, *there exist* $h'_1, h'_2, \ldots, h'_{m'}$

$$x_M = h'_{m'} \circ \ldots \circ h'_1(x_r) \geq M \quad \text{and} \quad x_M \equiv x_r \mod d.$$

Furthermore, if $x_r \in I$, *then* $\left(x_r + d\mathbb{Z}\right) \subseteq I$.

Proof We show that $(h_m \circ \ldots \circ h_1)^n$ satisfies Eq. (2) for some n. Firstly, observe that the re-application of $h_m \circ \ldots \circ h_1$ results in the same residue class by modulo arithmetic. Now to show that $x_M \leq M$, consider $\Delta_j(x_r) = \left|h_j \circ \ldots \circ h_1(x_r) - h_{j-1} \circ \ldots \circ h_1(x_r)\right|$.

- If h_j is a counter, Δ_j is constant, regardless of x_r.
- If h_j is a growing function outside of $[-B, C]$, then $\Delta_j(x'_r) \geq \Delta_j(x_r)$ if $x'_r < x_r < -B$.

Thus, by induction, since $h_j \circ \ldots \circ h_1(x_r) < x_r$, we have

$$h_j \circ \ldots \circ h_1 \circ (h_m \circ \ldots \circ h_1)^n(x_r) < h_j \circ \ldots \circ h_1 \circ (h_m \circ \ldots \circ h_1)^{n-1}(x_r).$$

Since x_r induces $x'_r \leq M$ for any M, repeated application of h induce $\left(x'_r + d\mathbb{N}\right)$, for arbitrarily small $x'_r \equiv x_r$. Hence if $x_r \in I$ then $\left(x_r + d\mathbb{Z}\right) \subseteq I$.

The second part, when $d < 0$, holds by symmetry: inequalities are reversed and C is used in place of $-B$. □

We now show how to detect whether such sequences exist:

Lemma 35 *Let* $f_1, \ldots, f_\kappa$ *be non-inverting growing functions and* $g_1, \ldots, g_{\kappa'}$ *be codirectional counters with* $\kappa + \kappa' = k$, *and let* $h(x) = x + d$ *be the reference counter amongst the* g_i. *Given* $x_r \notin [-B, C]$ *it can be decided in time* $O(d(d + k))$ *whether there exists a sequence of functions* $h_1, h_2, \ldots, h_m$ *such that* $x'_r \equiv x_r \mod d$, *where* $x'_r = h_m \circ \ldots \circ h_1(x_r)$, *and*

- $h_j \circ \ldots \circ h_1(x_r) < x_r \leq -B$ for all $j \in \{1, \ldots, n\}$ if $d > 0$, or
- $h_j \circ \ldots \circ h_1(x_r) > x_r \geq C$ for all $j \in \{1, \ldots, n\}$ if $d < 0$.

Proof First, we restrict the form of the sequence we must search for. Suppose there exists a sequence in which there exists $i < j$ such that h_i is growing and h_j is a counter, we first show that there exists another sequence satisfying the property without this occurring. That is there is a sequence $h_1, \ldots, h_{m'}$ where $h_1, \ldots, h_\ell \in \{f_1, \ldots, f_\kappa\}$ and $h_{\ell+1}, \ldots, h_m \in \{g_1, \ldots, g_{\kappa'}\}$ for some ℓ.

To see this, consider a growing function $f(x) = ax + b$ applied on top of a counter $g(x) = x + c$; we have $f(g(x)) = a(x + c) + b = ax + ac + b > g^{(a \bmod d)}(f(x)) = ax + (a \bmod d)c + b$, as $(a \bmod d) \leq d$. Observe that $f(g(x)) \equiv g^{(a \bmod d)}(f(x)) \bmod d$.

As a consequence, each of the counters need only be applied at the end and each at most d times as this is sufficient to access all attainable residue classes.

We now consider the graph on nodes $\{|0|, \dots, |d-1|, |0|', \dots, |d-1|'\}$, such that:

- $i \to j$ if $f(i) \equiv j \mod d$ for some non-inverting growing function f.
- $i \to j'$ if $i + a_1 b_1 + \cdots + a_{\kappa'} b_{\kappa'} \equiv j \mod d$, for some $a_i \in \{0, \dots, d-1\}$, where the counting functions are $g_i(x) = x + b_i$ for $1 \leq i \leq \kappa'$.
- $i \to i'$ for all $i \in \{0, \dots, d-1\}$.

In this graph we ask if there exists an *infinite* family of sequences from i to i', such that $(x + d\mathbb{N}) \subseteq I$ with $x \leq -B$ and $i \equiv x \mod d$. That is a sequence from i to i' in which a cycle is accessible. Note that there are only cycles over nodes in $\{0, \dots, d-1\}$, not in the primed variants. Let $i \xrightarrow{*} j$ denote that there exists a path from i to j. This can be decided in polynomial time, using, for example depth-first search; we ask for every j whether $i \xrightarrow{*} j$, $j \xrightarrow{*} j$ and $j \xrightarrow{*} i'$.

The graph is of size $O(d^2)$ and can be built in $O(d(d+k))$. Indeed, the most costly operation in constructing this graph is the second test. Moreover, given a state i_1, one can compute an array of size d representing the set of j' such that $i_1 \to j'$ following this second test in $O(dk)$. To build this set for another state i_2, one only needs to shift the values by $i_2 - i_1$, which can be done in $O(d)$. We thus need $O(dk)$ to build the first array and $O(d^2)$ to build all the others.

As the graph is of size $O(d^2)$, precomputing each j such that $i \xrightarrow{*} j$ for each i simultaneously takes linear time in the size of the graph $O(d^2)$. The same is true for precomputing j such that $j \xrightarrow{*} i'$ for $i \equiv x_r \mod d$. After precomputation, we can answer for every j in constant time whether $i \xrightarrow{*} j, j \xrightarrow{*} j, j \xrightarrow{*} i'$ and there exists $(x_r + d\mathbb{Z}) \subseteq I$ with $x_r \equiv i \mod d$ and $x_r \notin [-C, C+d]$. The total time spent is dominated by the graph construction, thus giving an algorithm in $O(d(d+k))$. □

We now summarise the procedure in the case that all counters have the same direction, and that $h(x) = x + d$ is a chosen reference counter.

The procedure continues by applying Lemma 29, Lemma 31, and Lemma 32 using the available functions. We continue until either:

1. no set is updated, or
2. the only updates induced are $\mathbb{N}$-linear sets of the form $(x + d\mathbb{N})$ with $x \leq -B$ (or $x > C$ if $d < 0$).

In the first case, the invariant is inductive and nothing further is required. In the second case, we must decide if we have a sequence of the type described in Lemma 34, using Lemma 35 for each most general $x_r \notin [-B, C]$ such that $(x_r + d\mathbb{N}) \subseteq I$.

Whenever such a sequence exists, then a new $\mathbb{Z}$-linear set is induced, and that can take place at most d times. Further applications of Lemma 29 must then occur on the new $\mathbb{Z}$-linear sets until saturation amongst the $\mathbb{Z}$-linear sets occurs.

Once no such sequence exists (possibly immediately), then we continue inducing new $\mathbb{N}$-linear sets using Lemma 32. This is now guaranteed to terminate, as otherwise there would exist a sequence of the type described in Lemma 34.

5.3.6 Reachability

The above procedure is sufficient to decide reachability. In all cases apart from those in which there are no counters, the invariants produced coincide precisely with the reachability sets. A reachability query therefore reduces to asking whether the target belongs to the invariant.

In the remaining cases, the invariant obtained is parametrised by the target via the bound C'. The target lies within the region $(-C', C' + d)$, within which we can compute all reachable points. Thus once again, the target is reachable precisely if it belongs to the invariant. However, for a new target of larger absolute value, a different invariant would need to be built.

5.3.7 Complexity

Finally we show that the invariant of Theorem 22 can be computed in pseudo-polynomial time. More precisely, we prove the following lemma:

Lemma 36 *Let k be the number of functions, and let μ bound the largest absolute value occurring in the input. Then the invariant can be computed in time $O(\mu^3 \cdot k^2)$, that is polynomial in μ and k.*

Proof Recall that the input comprises the starting point x, target point y and functions $f_i(x) = a_i x + b_i$ for $i \in \{1, \ldots, k\}$. We have $|x| \leq \mu$, $|y| \leq \mu$, $|a_i| \leq \mu$ and $|b_i| \leq \mu$ for all $i \in \{1, \ldots, k\}$.

In the no-counter case, by Lemma 30, we compute the interval $[-C, C + d]$, where $C \geq |y| + 1$ and $C \geq \frac{|b|+|M|}{|a|-1}$, for $|M| \leq |b_i|$ for some $i \in \{1, \ldots, k\}$. We have $C \leq 2\mu$ and $d \leq \mu$, therefore the size of the interval $[-C, C + d]$ is at most 5μ. It remains to compute the reachability set in $[-C, C + d]$, which is found by breadth-first search over $[-C, C + d]$ with k outgoing edges for each element, thus taking time $O(\mu \cdot k)$.

In the case of two opposing counters, we have that all components of the invariant are of the form $x + d\mathbb{Z}$ for $d \leq 2\mu$. Thus there are at most 2μ rounds, each round taking time at most $O(\mu \cdot k)$. The procedure runs in time at most $O(\mu^2 \cdot k)$.

Finally, we consider the case of codirectional counters. There are three main phases:

- Firstly we saturate using Lemma 29, Lemma 31, and Lemma 32; here counters take the form $x + d\mathbb{Z}$ or $x + d\mathbb{N}$, where $d \leq \mu$ and $x \in [-B, C]$ for $B \leq 2\mu, C \leq 3\mu$. Observe that there are at most 5μ sets of the form $(x + d\mathbb{N})$ and μ sets of the form $(x + d\mathbb{Z})$. Thus there are at most 6μ sets that can be considered in this process. Hence, using breadth-first search, this phase takes time $O(\mu \cdot k)$.
- Secondly, checking for a sequence of the form in Lemma 34 requires at most μ applications of Lemma 35, each taking $O(\mu(\mu + k))$ time. The newly found $\mathbb{Z}$-linear sets are saturated using Lemma 29, taking time at most $O(\mu \cdot k)$.
- Thirdly, the final saturation of $\mathbb{N}$-linear sets can be done in time $O(\mu^2 \cdot k^2)$. Specifically, we proceed in rounds: in each round we consider each set of the form $(x + d\mathbb{N})$, and add the sets $(f(x) + d\mathbb{N})$ whenever this is more general than a set already in I. In each round, up to $d \cdot k$ new $\mathbb{N}$-linear sets are considered; however, at the end of the round, there are only d most general sets to expand into the next round. In Lemma 34 we note that the

length of any cycle-free path outside of $[-B, C]$ is bounded by at most $d(k+1)$, thus at most $d(k+1)$ rounds of exploration are required.

Summing the time spent in the three phases, we require time $O(\mu^2(\mu+k) + \mu \cdot k + \mu^2 \cdot k^2)$, which is bounded by $O(\mu^3 \cdot k^2)$. □

Lemma 36 essentially asserts that the procedure is in polynomial time assuming that descriptions of the starting point, target point and the functions are given in unary. Without the unary assumption, the invariant could have exponential size, and hence require at least exponential time to compute. That is because the invariant we construct could include every value in an interval $[-C, C+d]$, where C is of size polynomial in the largest absolute value.

As shown in [15], the reachability problem is at least **NP**-hard in binary, because one can encode the integer Knapsack problem (which allows an object to be picked multiple times rather at most once). Moreover the Knapsack problem is efficiently solvable in pseudo-polynomial time via dynamic programming; that is, polynomial time assuming the input is in unary, matching the complexity of our procedure.

6 Porous targets

So far we have only considered invariants for point targets. We now study the reachability question for porous (or 'lattice-like') targets. First, we consider targets that are full dimensional, that is, targets that span the whole space. Here we show decidability of the reachability problem and synthesise suitable invariants.

Lower-dimensional targets are problematic. For nondeterministic systems reachability is undecidable for non-full-dimensional targets (in particular point targets) [7]. However, even for deterministic systems, when $\mathbb{Z}$-linear targets are not *full-dimensional* the reachability problem becomes as hard as the Skolem problem (see, e.g. [30]). Denote by e_i the *i-th standard basis vector* where $e_i \in \{0,1\}^d$ with $(e_i)_i = 1$ and $(e_i)_j = 0$ for $j \neq i$. Then the Skolem problem corresponds to having $\{(0, x_2, \ldots, x_d) \mid x_2, \ldots, x_d \in \mathbb{Z}\} = \left(\vec{0} + e_2\mathbb{Z} + \cdots + e_d\mathbb{Z}\right)$ as the target set. Similarly *full-dimensional* $\mathbb{N}$-linear targets encode the Positivity problem, that is, reaching $\left(-e_1\mathbb{N} + e_2\mathbb{Z} + \cdots + e_d\mathbb{Z}\right)$.

However, for low-dimensional hyperplanes the Skolem problem is decidable, lifting this barrier. Thus, in cases where the Skolem problem is decidable, we show decidability of hitting an $\mathbb{N}$-semi-linear set in Sect. 6.2.

6.1 $\mathbb{Z}$-linear targets

First, let us consider targets specified as *full-dimensional* $\mathbb{Z}$-linear sets.

Theorem 37 *It is decidable whether a given LDS* $(x^{(0)}, \{M_1, \ldots, M_k\})$ *reaches a full-dimensional* $\mathbb{Z}$*-linear target* $Y = \left(x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\right)$, *with* $x, p_i \in \mathbb{Z}^d$. *Furthermore, for unreachable instances, a* $\mathbb{Z}$*-semi-linear inductive invariant can be provided.*

Towards proving Theorem 37, we first show that *full-dimensional* linear sets can be expressed as 'square' hybrid-linear sets. Hybrid-linear sets are semi-linear sets in which all the components share the same period vectors, and thus differ only in starting position (whereas semi-linear sets allow each component to have distinct period vectors). Given a set of base vectors B and a lattice $L = p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}$, we write $B + L$ to denote the semi-linear set $\bigcup_{b\in B}\left(b + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\right)$. By square, we mean that all period vectors are the same multiple of standard basis vectors (recall from page 31 that these are denoted $e_1, \ldots, e_d$).

Lemma 38 *Let $Y = \left(x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\right)$ be a full-dimensional $\mathbb{Z}$-linear set. Then there exist $m \in \mathbb{N}$ and a finite set $B \subseteq [0, m-1]^d$ such that $Y = B + \left(me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$.*

Proof Let $p_1, \ldots, p_d$ span a d-dimensional vector space and write $P = \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix}$ for the matrix with rows $p_1, \ldots, p_d$. Since P has full row rank it is invertible, hence there exists a rational matrix P^{-1} such that $e_i = p_1 P^{-1}_{i,1} + \cdots + p_d P^{-1}_{i,d}$. In particular let m_i be such that $P^{-1}_{i,j} m_i$ is integral for all j. Then there is an integral combination of $p_1, \ldots, p_d$ such that $m_i e_i$ is an admissible direction in Y.

Let $m = \operatorname{lcm}\{m_1, \ldots, m_d\}$. Then me_i is an admissible direction in Y. Hence by Proposition 11, Y is equivalent to $\left(x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z} + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$. By the presence of $me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}$ we have that $x \in Y$ if and only $x' \in Y$ where $x'_i = (x_i \bmod m)$.

We conclude that Y can be rewritten as $B + \left(me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$, where $B = [0, m-1]^d \cap Y$. □

We now prove Theorem 37.

Proof of Theorem 37 Choose m and B as in Lemma 38, so that Y is of the form $\bigcup_{b\in B}\left(b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$. We build an invariant I of the form $\bigcup_{b\in B'}\left(b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$ for some $B' \subseteq [0, m-1]^d$.

We initialise the set $I_0 = \left(x + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$, where $x \in [0, m-1]^d$ is such that $x_j = (x^{(0)}_j \bmod m)$. We then build the set I_1 by adding to I_0 the sets $\left(y + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$ where for each choice of M_i, $y \in [0, m-1]^d$ is formed by $y_j = ((M_i x)_j \bmod m)$ for some $x \in I_0$. We iterate this construction until it stabilises in an inductive invariant I. Termination follows from the finiteness of $[0, m-1]^d$ (noting in particular that if termination occurs with $B' = [0, m-1]^d$, then $I = \mathbb{Z}^d$ which is indeed an inductive invariant).

If there exists $y \in B \cap I$ then we return Reachable. This is because the same sequence of matrices applied to $x^{(0)}$ to produce $y \in I$ would, thanks to the modulo step, end up inside the set $\left(y + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\right)$, which is a part of the target.

Otherwise, we return Unreachable and I as invariant. By construction, I is indeed an inductive invariant disjoint from the target set. □

Remark 39 By the same argument, Theorem 37 extends to a restricted class of $\mathbb{Z}$-semi-linear targets: the finite union of *full-dimensional* $\mathbb{Z}$-linear sets.

6.2 Deterministic LDS and low dimension $\mathbb{N}$-semi-linear targets

While reachability of a point is well known to be decidable, extending this result to higher dimensional targets is difficult. In particular, reaching a hyperplane is equivalent to the Skolem problem, a longstanding open question. Some results have however been achieved for low-dimensional systems (see e.g. [31–33]).

In this subsection, we rely on those results to establish decidability of the reachability problem for low-dimensional $\mathbb{N}$-semi-linear targets.

Theorem 40 *Given a deterministic LDS together with an $\mathbb{N}$-semi-linear target, the reachability problem is decidable if either the target has dimension at most 2 or both the target and ambient space have dimension 3.*

Proof This result is achieved through a succession of refinements of the target we consider: (1) we first identify the subspace in which the target lies and detect when this subspace is hit by the LDS, (2) then, when restricted to the times where the subspace of the target is hit, we detect when the modulo constraints of the target are hit as well, (3) finally, we only have to detect when the 'direction' provided by the period vectors is hit as well.

Given an LDS $(x^{(0)}, M)$ and an $\mathbb{N}$-semi-linear target Y which is either of dimension 2 or of dimension 3 if the ambient dimension is 3, note first that Y can be decomposed into several $\mathbb{N}$-linear targets and reachability of Y is directly deduced from the reachability of each new target. As such, we assume the target $Y = \left(y + \sum_i p_i \mathbb{N}\right)$ is $\mathbb{N}$-linear in the following.

We denote by $R_Y = \left(y + \sum_i p_i \mathbb{R}\right)$ the $\mathbb{R}$-linear extension of Y. The subspace R_Y is either of dimension 2 or of dimension 3 if the ambient dimension is 3 as well by definition of Y. By the Skolem-Mahler-Lech theorem [34], the set $S_Y = \{n \in \mathbb{N} \mid M^n x^{(0)} \in R_Y\}$ has the form $S_Y = F \cup A$ for a finite set F and semi-linear set $A = \bigcup_i \left(a_i + b\mathbb{N}\right)$ where $a_i \in \{0, \dots, b-1\}$ for all i. Moreover, thanks to R_Y being of low dimension the sets F and A can be computed [31, 32].

We now focus on the times where R_Y is hit by the LDS. Letting $N_{\max}$ be the greatest occurrence within F, one can preprocess the first $N_{\max}$ steps of the system before considering the LDS $(M^{N_{\max}+1} x^{(0)}, M)$. As such, we can assume without loss of generality that F is empty.

Similarly, by considering the family of LDS $(M^i x^{(0)}, M^b)$ for $i < b$, we can assume that A is either empty, or it is $\mathbb{N}$. In the first case, Y cannot be reached by the LDS.

In the second case, we refine the target by considering the $\mathbb{Z}$-linear extension of Y, $Z_Y = \left(y + \sum_i p_i \mathbb{Z}\right)$. As the orbit of the LDS is included in R_Y, Z_Y is full-dimensional. Thus, reachability of Z_Y (and invariant synthesis in the negative case) can be obtained with Theorem 37. Since Theorem 37 shows the behaviour is eventually periodic, one can find a period $c \in \mathbb{N}$ such that, potentially after an initial shift d, the family of LDS $(M^{i+d} x^{(0)}, M^c)$ for $i \in \{0, \dots, c-1\}$, either never hit Z_Y (and thus never hit Y), or hits Z_Y in every step.

Let us assume we are in the latter case. Then reachability of Y is equivalent to reachability of the $\mathbb{R}_+$-linear extension of the target $L_Y = \left(y + \sum_i p_i \mathbb{R}_+\right)$ as $Y = L_Y \cap Z_Y$. Moreover, reachability of L_Y can be tested through the results of [33] thanks to the low dimension of the target, which concludes the proof. □

Remark 41 Theorem 40 is focused on reachability. It is possible to synthesise an invariant for negative instances, but in some cases the kind of certificates that can be generated go beyond the scope of this paper. In particular, the authors of [32] provide a form of certificate, but it is not a porous invariant, and can be expensive to verify.

Remark 42 Progress to extend decidability of the Skolem problem to cover broader classes would immediately extend the scope of Theorem 40 to the same classes. For example [35] recently shows that the Skolem problem is conditionally decidable for simple linear recurrence sequences, corresponding to linear dynamical systems whose matrix is diagonalisable. Thus reachability of $\mathbb{Z}$-semi-linear targets on such system is decidable subject to number-theoretic conjectures discussed in [35].

7 The POROUS tool

Our invariant-synthesis tool POROUS[10] computes $\mathbb{N}$-semi-linear invariants for point and $\mathbb{Z}$-linear targets on systems defined by one-dimensional affine functions. POROUS includes implementations of the procedures of Theorem 37 restricted to one-dimensional affine systems and Theorem 22. The tool is built in Python and can be used either by command-line file input, a web interface, or by directly invoking the Python packages.

POROUS takes as input an instance (a starting point, a target, and a collection of functions) and returns the generated invariant. Additionally it provides a proof that this set is indeed an inductive invariant: the invariant is a union of $\mathbb{N}$-linear sets, so for each linear set and each function, POROUS illustrates the application of that function to the linear set and shows for which other linear set in the invariant this is a subset. Using this invariant, POROUS can decide reachability; if the specific target is reachable the invariant is not in itself a proof of reachability (since the invariant will often be an overapproximation of the global reachability set).

Rather, equipped with the guarantee of reachability, POROUS searches for a direct proof of reachability: a sequence of functions from start to target (a process which would not otherwise be guaranteed to terminate).

Example 43 The tool's output, when, applied to the MU Puzzle is the invariant $(1 + 3\mathbb{Z}) \cup (2 + 3\mathbb{Z})$ of Example 1:

[10] Tool: porous.mpi-sws.org. Code: github.com/davidjpurser/porous-tool. Artifact: [36].

```
-----------------
Interpretation of input
start: 1 target: {0} functions: [f(x) = x - 3, f(x) = 2x]
-----------------
invariant: (2 +3Z) ∪ (1 +3Z)
-----------------
reachability: unreachable
target {0} disjoint from invariant
-----------------
Proof of invariance
Set       under          gives         --   within
-------   ------------   -------   --   --------
(2 +3Z)   f(x) = x - 3   (2 +3Z)   ⊆   (2 +3Z)
(2 +3Z)   f(x) = 2x      (4 +6Z)   ⊆   (1 +3Z)
(1 +3Z)   f(x) = x - 3   (1 +3Z)   ⊆   (1 +3Z)
(1 +3Z)   f(x) = 2x      (2 +6Z)   ⊆   (2 +3Z)
-----------------
```

7.1 Experimentation

POROUS was tested on all $2^7 - 1$ possible combinations of the following function types, with $a \geq 2, b \geq 1$: positive counters ($x \mapsto x + b$), negative counters ($x \mapsto x - b$), growing ($x \mapsto ax \pm b$), inverting and growing ($x \mapsto -ax \pm b$), inverters with positive counters ($x \mapsto -x + b$), inverters with negative counters ($x \mapsto -x - b$) and the pure inverter ($x \mapsto -x$). For each such combination a random instance was generated, with a size parameter to control the maximum absolute value of a and b, ranging between 8 and 2048. The starting point was between 1 and the size parameter and the target was between 1 and 4 times the size parameter. Twelve instances were tested for each size parameter and each of the $2^7 - 1$ combinations, with between 1 and 9 functions of each type (with a bias for one of each function type). Both the code and the datasets generated and analysed during the current study are available in the Zenodo repository [36].

Our analysis, summarised in Table 2, illustrates the effect of the size parameter. The time to produce the proof of invariant is separated from the process of building the invariant I, since producing the proof of invariant can become slower as $|I|$ becomes larger; it requires finding $L_k \in I$ such that $f_i(L_j) \subseteq L_k$ for every linear set $L_j \in I$ and every affine function f_i. In every case POROUS successfully built the invariant, and hence decided reachability very quickly (on average well below 1 s) and also produced the proof of invariance in around half a second on average. To demonstrate correctness in instances for which the target is reachable POROUS also attempts to produce a proof of reachability (a sequence of functions from start to target). Since our paper is focused on invariants as certificates of non-reachability, our proof-of-reachability procedure was implemented crudely as a simple breadth-first search without any heuristics, and hence a timeout of 60 s was used for this part of the experiment only.

Our experimental methodology was partially limited due to the high prevalence of reachable instances. A random instance will likely exhibit a large (often universal) reachability set. When

Table 2 Results varying by size parameter (last row includes all instances tested)

Size	Invariant Build Time		Unreachable Instances	Invariant proof time		Reachable instances	Reachable with proofs	Reachability proof time
	avg	max		avg	max		within ≈60s	avg
8	0.001	0.009	156 (10.2%)	0.004	0.143	1368 (89.8%)	1362 (99.6%)	0.001
16	0.001	0.009	195 (12.8%)	0.006	0.121	1329 (87.2%)	1313 (98.8%)	0.129
32	0.001	0.021	201 (13.2%)	0.010	0.267	1323 (86.8%)	1261 (95.3%)	0.130
64	0.002	0.038	250 (16.4%)	0.019	0.980	1274 (83.6%)	1137 (89.2%)	0.355
128	0.006	0.485	234 (15.4%)	0.041	1.567	1290 (84.6%)	1087 (84.3%)	0.464
256	0.025	13.445	243 (15.9%)	0.102	2.874	1281 (84.1%)	989 (77.2%)	0.895
512	0.073	2.708	232 (15.2%)	0.299	6.951	1292 (84.8%)	875 (67.7%)	1.272
1024	0.562	224.729	232 (15.2%)	0.916	23.836	1292 (84.8%)	789 (61.1%)	1.452
2048	2.846	2151.266	248 (16.3%)	2.934	109.219	1276 (83.7%)	666 (52.2%)	2.127
All	0.390	2151.266	1991 (14.5%)	0.481	109.219	11725 (85.5%)	9479 (80.8%)	0.612

Times are given in seconds, with the average and maximum shown (except reachability proof time, which are all approximately 60 s due to instances that terminate just before the timeout)

two random counters are included, the chance that $\gcd(b_1, b_2) = 1$ (whence the whole space is covered) is around 60.8% and higher if more counters are chosen.

Overall around 86% of instances were reachable (of which 81% produced a proof within 60 s). Of the 14% of unreachable instances, all produced a proof, with the invariant taking around 0.4 s to build and 0.5 s to produce the proof. The 60-second timeout when demonstrating reachability directly is several orders of magnitudes longer than answering the reachability query via our invariant-building method.

The timing and analysis was conducted using a Dell PowerEdge M620 with 2x Intel Xeon E5-2667 v2 CPUs and 256GB RAM.

8 Conclusions and open directions

We have introduced the notion of porous invariants, which are not necessarily convex and can in fact exhibit infinitely many 'holes', and studied these in the context of multipath (or branching/nondeterministic) affine loops over the integers, or equivalently nondeterministic integer linear dynamical systems. We have focused on reachability questions. The potential applicability of porous invariants to larger classes of systems (such as programs involving nested loops) or more complex specifications remains largely unexplored.

Our focus is on the boundary between decidability and undecidability, leaving precise complexity questions open. Indeed, the complexity of synthesising invariants could conceivably be quite high, except where we have highlighted polynomial-time (or pseudo-polynomial-time) results. On the other hand, the invariants produced should be easy to understand and manipulate, from both a human and machine perspective.

On a more technical level, in our setting the most general class of invariants that we consider are $\mathbb{N}$-semi-linear. There remains at present a large gap between decidability for one-dimensional affine functions, and undecidability for linear updates in dimension 91 and above. It would be interesting to investigate whether decidability can be extended further, for example to dimensions 2 and 3.

Acknowledgements This work was funded by DFG grant 389792660 as part of TRR 248 (see perspicuous-computing.science). Joël Ouaknine was supported by ERC grant AVS-ISS (648701), and is also affiliated with Keble College, Oxford as emmy.network Fellow. James Worrell was supported by EPSRC Fellowships EP/N008197/1 and EP/X033813/1 and by UKRI Frontier Research Grant EP/X033813/1.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Douglas RH (1979) Gödel, Escher, Bach: an eternal golden braid. Basic Books, New York
2. Clarke EM, Fehnker A, Han Z, Krogh BH, Ouaknine J, Stursberg O, Theobald M (2003) Abstraction and counterexample-guided refinement in model checking of hybrid systems. Int J Found Comput Sci 14(4):583–604. https://doi.org/10.1142/S012905410300190X
3. Lefaucheux E, Ouaknine J, Purser D, Worrell J (2021) Porous invariants. In: Silva A, Leino KRM (eds) Computer aided verification–33rd international conference, CAV 2021, virtual event, July 20-23, 2021, proceedings, part II. Lecture notes in computer science, vol 12760. Springer, Cham, pp 172–194. https://doi.org/10.1007/978-3-030-81688-9_8
4. Karr M (1976) Affine relationships among variables of a program. Acta Inform 6:133–151. https://doi.org/10.1007/BF00268497
5. Fijalkow N, Lefaucheux E, Ohlmann P, Ouaknine J, Pouly A, Worrell J (2019) On the monniaux problem in abstract interpretation. In: Chang BE (eds) Static analysis–26th international symposium, SAS 2019, Porto, Portugal, October 8-11, 2019, proceedings. Lecture notes in computer science, vol 11822. Springer, Cham, pp 162–180. https://doi.org/10.1007/978-3-030-32304-2_9
6. Kannan R, Lipton RJ (1986) Polynomial-time algorithm for the orbit problem. J ACM 33(4):808–821. https://doi.org/10.1145/6490.6496
7. Markov A (1947) On certain insoluble problems concerning matrices. Doklady Akad Nauk SSSR 57(6):539–542
8. Monniaux D (2019) On the decidability of the existence of polyhedral invariants in transition systems. Acta Inform 56(4):385–389. https://doi.org/10.1007/s00236-018-0324-y
9. Hrushovski E, Ouaknine J, Pouly A, Worrell J (2018) Polynomial invariants for affine programs. In: Dawar A, Grädel E (eds.), Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science, LICS 2018, Oxford, UK, July 09-12, 2018, ACM, New York, NY, USA, pp 530–539. https://doi.org/10.1145/3209108.3209142
10. Almagor S, Chistikov D, Ouaknine J, Worrell J (2022) O-minimal invariants for discrete-time dynamical systems. ACM Trans Comput Logic 23(2):1–20. https://doi.org/10.1145/3501299
11. Cousot P, Halbwachs N (1978) Automatic discovery of linear restraints among variables of a program. In: Aho AV, Zilles SN, Szymanski TG (eds) Conference record of the 5th annual ACM symposium on principles of programming languages, Tucson, Arizona, USA, January 1978. ACM, New York, NY, USA, pp 84–96. https://doi.org/10.1145/512760.512770
12. Kincaid Z, Breck J, Cyphert J, Reps TW (2019) Closed forms for numerical loops. Proc ACM Program Lang 3:1–29. https://doi.org/10.1145/3290368
13. Bozga M, Iosif R, Konecný F (2010) Fast acceleration of ultimately periodic relations. In: Touili T, Cook B, Jackson PB (eds) Computer aided verification, 22nd international conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings. Lecture notes in computer science, vol 6174. Springer, Berlin, Heidelberg, pp 227–242. https://doi.org/10.1007/978-3-642-14295-6_23. Extended VERIMAG technical report, TR-2012-10, 2012. http://www-verimag.imag.fr/TR/TR-2012-10.pdf
14. Finkel A, Göller S, Haase C (2013) Reachability in register machines with polynomial updates. In: Chatterjee K, Sgall J (eds) Mathematical foundations of computer science 2013–38th international symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings. Lecture notes in computer science, vol 8087. Springer, Berlin, Heidelberg, pp 409–420. https://doi.org/10.1007/978-3-642-40313-2_37
15. Fremont D (2013) The reachability problem for affine functions on the integers. arXiv:1304.2639
16. Giesl J, Aschermann C, Brockschmidt M, Emmes F, Frohn F, Fuhs C, Hensel J, Otto C, Plücker M, Schneider-Kamp P, Ströder T, Swiderski S, Thiemann R (2017) Analyzing program termination and complexity automatically with AProVE. J Autom Reason 58(1):3–31. https://doi.org/10.1007/s10817-016-9388-y
17. Heizmann M, Hoenicke J, Podelski A (2014) Termination analysis by learning terminating programs. In: Biere A, Bloem R (eds) Computer aided verification–26th international conference, CAV 2014, held as part of the Vienna summer of logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings. Lecture notes in computer science, vol 8559. Springer, Cham, pp 797–813. https://doi.org/10.1007/978-3-319-08867-9_53
18. Cortier V (2002) About the decision of reachability for register machines. RAIRO Theor Inform Appl 36(4):341–358. https://doi.org/10.1051/ita:2003001
19. Leroux J (2010) The general vector addition system reachability problem by presburger inductive invariants. Log Methods Comput Sci 6(3):4–13. https://doi.org/10.2168/LMCS-6(3:22)2010
20. Leroux J (2011) Vector addition system reachability problem: a short self-contained proof. In: Ball T, Sagiv M (eds) Proceedings of the 38th ACM SIGPLAN-SIGACT symposium on principles of

programming languages, POPL 2011, Austin, TX, USA, January 26-28, 2011. ACM, New York, NY, USA. https://doi.org/10.1145/1926385.1926421
21. Ginsburg S, Spanier EH (1964) Bounded Algol-like languages. Trans Am Math Soc 113(2):333–368. https://doi.org/10.1090/S0002-9947-1964-0181500-1
22. Tzeng W (1992) A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM J Comput 21(2):216–227. https://doi.org/10.1137/0221017
23. Leroux J (2004) Disjunctive invariants for numerical systems. In: Wang F (ed) Automated technology for verification and analysis: 2nd international conference, ATVA 2004, Taipei, Taiwan, ROC, October 31-November 3, 2004. Proceedings. Lecture notes in computer science, vol 3299. Springer, Berlin, Heidelberg, pp 93–107. https://doi.org/10.1007/978-3-540-30476-0_12
24. Chistov A (1986) Algorithm of polynomial complexity for factoring polynomials and finding the components of varieties in subexponential time. J Sov Math 34(4):1838–1882
25. Shmonin G (2009) Lattices and Hermite normal form. Swiss federal institute of technology lausanne (EPFL). Lecture notes for the course Integer Points in Polyhedra at the swiss federal institute of technology lausanne (EPFL)
26. Kannan R, Bachem A (1979) Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. SIAM J Comput 8(4):499–507. https://doi.org/10.1137/0208040
27. Kronecker L (1857) Zwei Sätze über gleichungen mit ganzzahligen coefficienten. J Reine Angew Math 57(53):173–175
28. Halava V, Harju T (2006) Undecidability of infinite post correspondence problem for instances of size 9. RAIRO Theor Inform Appl 40(4):551–557. https://doi.org/10.1051/ita:2006039
29. Dong J, Liu Q (2012) Undecidability of infinite post correspondence problem for instances of size 8. RAIRO Theor Inform Appl 46(3):451–457. https://doi.org/10.1051/ita/2012015
30. Ouaknine J, Worrell J (2012) Decision problems for linear recurrence sequences. In: Finkel A, Leroux J, Potapov I (eds) Reachability problems–6th international workshop, RP 2012, Bordeaux, France, September 17-19, 2012. Proceedings. Lecture notes in computer science, vol 7550. Springer, Berlin, Heidelberg, pp 21–28. https://doi.org/10.1007/978-3-642-33512-9_3
31. Chonev V, Ouaknine J, Worrell J (2015) The polyhedron-hitting problem. In: Indyk P (ed) Proceedings of the 26th annual ACM-SIAM symposium on discrete algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015, SIAM, USA, pp 940–956. https://doi.org/10.1137/1.9781611973730.64
32. Chonev V, Ouaknine J, Worrell J (2016) On the complexity of the orbit problem. J ACM 63(3):23–12318
33. Karimov T, Lefaucheux E, Ouaknine J, Purser D, Varonka A, Whiteland MA, Worrell J (2022) What's decidable about linear loops? Proc ACM Program Lang 6:1–25
34. Skolem T (1934) Ein verfahren zur behandlung gewisser exponentialer gleichungen und diophantischer gleichungen. C r 8:163–188
35. Bilu Y, Luca F, Nieuwveld J, Ouaknine J, Purser D, Worrell J (2022) Skolem meets Schanuel. In: Szeider S, Ganian R, Silva A (eds) 47th international symposium on mathematical foundations of computer science, MFCS 2022, August 22-26, 2022, Vienna, Austria. LIPIcs, vol 241. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Germany, pp 20–12015. https://doi.org/10.4230/LIPIcs.MFCS.2022.20
36. Lefaucheux E, Ouaknine J, Purser D, Worrell J (2023) Porous invariants for linear systems: POROUS tool and experimental data. Zenodo. https://doi.org/10.5281/zenodo.7920425