Porous Invariants

Engel Lefaucheux[1], Joël Ouaknine[1], David Purser[1(✉)], and James Worrell[2]

[1] Max Planck Institute for Software Systems, Saarland Informatics Campus, Saarbrücken, Germany
dpurser@mpi-sws.org

[2] Department of Computer Science, Oxford University, Oxford, UK

Abstract. We introduce the notion of *porous invariants* for multipath (or branching/nondeterministic) affine loops over the integers; these invariants are not necessarily convex, and can in fact contain infinitely many 'holes'. Nevertheless, we show that in many cases such invariants can be automatically synthesised, and moreover can be used to settle (non-)reachability questions for various interesting classes of affine loops and target sets.

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 $(x^{(i)})$ 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 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. One requires some sort of 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 full version of this paper is available at http://arxiv.org/abs/2106.00662.

A. Silva and K. R. M. Leino (Eds.): CAV 2021, LNCS 12760, pp. 172–194, 2021.
https://doi.org/10.1007/978-3-030-81688-9_8

Meta Problem 1. *Consider a system with update functions $f_1, \ldots, f_n$. A set I is an inductive invariant if $f_i(I) \subseteq I$ for all i. Given a reachability query (x, Y) we search for a separating inductive invariant I such that $x \in I$ and $Y \cap I = \emptyset$.*

Meta Problem 1 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 a particular invariant and target domain, a reachability query has 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. For some classes of systems, it is known that distinguishing reachability (1) from unreachability (2, 3) is undecidable; it can also happen that determining whether a separating invariant exists (i.e., distinguishing (2) from (3)) is undecidable.

We note that the existence of *strongest* inductive invariants[1] is a desirable property for an invariant domain—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 state or not; if so, then you are done, and if not, no other invariant (from that class) can possibly do the trick either. However, unless (3) is excluded, computing the strongest invariant does not necessarily imply that reachability is decidable. Unfortunately, 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[2]. 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 $\{b + p_1\mathbb{N} + \cdots + p_m\mathbb{N}\}$ (by which we mean $\{b + a_1p_1 + \cdots + a_mp_m \mid a_1, \ldots, a_m \in \mathbb{N}\}$, see Definition 2).

Let us first consider a motivating example:

Example 1 (Hofstadter's MU Puzzle [7]). 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.

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

[1] Given two invariants I and I', we say that I is *stronger* than I' iff $I \subseteq I'$; thus *strongest* invariants correspond to *smallest* invariant sets.

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

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,[3] or a two-dimensional system with two linear updates.[4] The set $\{1+3\mathbb{Z}\} \cup \{2+3\mathbb{Z}\}$ is an inductive invariant, and we wish to synthesise this. (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 $\{2+3\mathbb{Z}\} \mapsto \{4+6\mathbb{Z}\} \subseteq \{1+3\mathbb{Z}\}$.)

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 above loop can never halt, regardless of the nondeterministic choices made.

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

The MU Puzzle was presented as a challenge for algorithmic verification in [4]; 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 $x \mapsto 2x \parallel x \mapsto x-3$).

Main Contributions. Our focus is on the automatic generation of porous invariants for multipath affine loops over the integers, or equivalently nondeterministic integer linear dynamical systems.

- 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 consider $\mathbb{R}, \mathbb{R}_+$-(semi)-linear sets, where we complete 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 (Theorem 2). These invariants may or may not separate the target under consideration.
  - If a $\mathbb{Z}$-linear invariant is not separating, we may instead look for an $\mathbb{N}$-semi-linear invariant (which generalises both $\mathbb{Z}$-semi-linear and $\mathbb{N}$-linear invariants), and we show that such an invariant can always be found

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

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

Table 1. Results for integer linear dynamical systems for a point target. Det/Non refers to deterministic or nondeterministic LDS. "Subsumed by ..." means that sufficient invariants can be generated, but of a more general type.

Dom	D/N	Linear	Semi-linear (SL)
$\mathbb{Z}$	Det	Strongest computable (Theorem 2)	No strongest (Sect. 4.1); subsumed by $\mathbb{N}$-SL
$\mathbb{Z}$	Non	Strongest computable (Theorem 2)	No strongest (Sect. 4.1)
$\mathbb{N}$	Det	No strongest (Sect. 4.1); subsumed by $\mathbb{N}$-SL	No strongest (Sect. 4.1), but sufficient computable (Theorem 4)
$\mathbb{N}$	Non	No strongest (Sect. 4.1)	1d-affine decidable (Theorem 6); undec. in general (Theorem 5)
$\mathbb{R}$	Det	Strongest: affine relations by Karr [17]	Strongest: affine closure on Zariski closure (Theorem 1)
$\mathbb{R}$	Non	Strongest: affine relations by Karr [17]	Strongest: affine closure on Zariski closure (Theorem 1)
$\mathbb{R}_+$	Det	No strongest (Sect. 4.1); subsumed by $\mathbb{R}_+$-SL	No strongest, but sufficient computable [8]
$\mathbb{R}_+$	Non	No strongest (Sect. 4.1)	Undecidable [8]

for any unreachable point target when dealing with *deterministic* integer linear dynamical systems (Theorem 4).

- However, for nondeterministic integer linear dynamical systems, computing an $\mathbb{N}$-semi-linear invariants is an undecidable problem in arbitrary dimension (Theorem 5). Nevertheless we show how such invariants can be constructed in a low-dimensional setting, in particular for affine updates in one dimension (Theorem 6). 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.

– 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 3). If the target is *not* full-dimensional then the reachability problem is Skolem-hard and undecidable for deterministic and nondeterministic systems respectively.
– In Sect. 6 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.

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 [16]. However for nondeterministic systems (where the update matrix is chosen nondeterministically from a finite set at each time step), reachability is undecidable, by reduction from the matrix semigroup membership problem [22].

In particular this entails that for unreachable nondeterministic instances we cannot hope *always* to be able to compute a separating invariant. In some cases

[5] The affine span covers the entire space.

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., polyhedral invariants [8]).

Various types of invariants have been studied for linear dynamical systems, including polyhedra [8,23], algebraic [15], and o-minimal [1] invariants. For certain classes of invariants (e.g., algebraic [15]), it is decidable whether a separating invariant exists, notwithstanding the reachability problem being undecidable. Other works (e.g., [5]) use heuristic approaches to generate invariants, without aiming for any sort of completeness.

Kincaid, Breck, Cyphert and Reps [18] study loops with linear updates, studying 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 [2] 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 [9], extending Fremont [10], show that reachability in a single dimension is **PSPACE**-complete for polynomial update functions (and allowing states can be used to control the sequences of updates which 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.

Other tools, e.g., AProVE [11] and Büchi Automizer [14] may (dis-)prove termination/reachability on *all* branches, but may not be able to prove termination/reachability on *some* branch.

Inductive invariants specified in Presburger arithmetic have been used to disprove reachability in vector addition systems [20]. A generalisation, 'almost semi-linear sets' [21] are also non-convex and 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 integers and $\mathbb{N}$ the 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 $(x \mod d)$ for the number in $\{0, \ldots, d-1\}$ such that $(x \mod d) \equiv x \mod d$.

Definition 1 (Integer Linear Dynamical Systems). *A d-dimensional integer linear dynamical system (LDS)* $(x^{(0)}, \{M_1, \ldots, M_k\})$ *is defined by an initial point* $x^{(0)} \in \mathbb{Z}^d$ *and a set of integer matrices* $M_1, \ldots, M_k \subseteq \mathbb{Z}^{d \times d}$. *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* $B_1, \ldots, B_m$ *such that* $B_1 \cdots B_m x^{(0)} = y$ *and* $B_i \in \{M_1, \ldots, M_k\}$ *for all* $1 \leq i \leq m$.

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

Definition 2 ($\mathbb{K}$-(semi)-linear sets). *A* linear set L *is defined by a base vector* $b \in \mathbb{Z}^d$ *and period vectors* $p_1, \ldots, p_d \in \mathbb{Z}^d$ *such that*

$$L = \{b + a_1 p_1 + \cdots + a_d p_d \mid a_1, \ldots, a_d \in \mathbb{K}\}.$$

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

$\mathbb{N}$-semi-linear sets are precisely those definable in Presburger arithmetic ($\mathrm{FO}(\mathbb{Z}, +, \leq)$) [12]. However, we can also consider $\mathbb{Z}$-semi-linear sets (corresponding to $\mathrm{FO}(\mathbb{Z}, +)$ without order), and the real counterparts ($\mathbb{R}$ and $\mathbb{R}_+$). Note that even if $\mathbb{K} = \mathbb{N}$ we still allow $p_i \in \mathbb{Z}^d$.

Definition 3. *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* $i \in \{1, \ldots, k\}$.

Note in particular that every inductive invariant contains the reachability set ($\mathcal{O} \subseteq I$). We are interested in the following problem:

Definition 4 (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*?*

In our setting, we are interested in classes $\mathcal{D}$ of 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 decision problem is, in theory, sufficient—although all of our proofs are constructive.

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., described as $\mathbb{R}$-(semi)-linear sets.

Strongest $\mathbb{R}$-linear invariants are given precisely by the affine hull of the reachability set, and can be computed using Karr's algorithm [17]. Moreover, we will show that strongest $\mathbb{R}$-semi-linear invariants also exist and can be computed by combining techniques for algebraic invariants [15] and $\mathbb{R}$-linear invariants.

$\mathbb{R}$-*linear.* Recall that a set L is $\mathbb{R}$-linear if $L = \{v_0 + v_1\mathbb{R} + \cdots + v_t\mathbb{R}\}$ 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

$$\overline{S}^a = \left\{ \sum_{i=1}^{k} \lambda_i x_i \mid k \in \mathbb{N}, x_i \in S, \lambda_i \in \mathbb{R}, \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

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 $\overline{\mathcal{O}}^a$ is precisely the strongest $\mathbb{R}$-linear invariant. Karr's algorithm [17,26] can be used to compute this strongest invariant in polynomial time. The next lemma follows from Theorem 3.1 of [26].

Lemma 1. *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 ($\overline{R_0}^a = \overline{\mathcal{O}}^a$),*
3. *the entries of the vectors in R_0 have absolute value at most $\mu_0 := (d\mu)^d$.*

Let $R_0 = \left\{x^{(0)}, r_1, \ldots, r_{d'}\right\}$ be obtained as per Lemma 1, with $d' \leq d$. The $\mathbb{R}$-linear invariant of the LDS is the affine span $\overline{R_0}^a$, which can be written as the $\mathbb{R}$-linear set $L_0 = \left\{x^{(0)} + (r_1 - x^{(0)})\mathbb{R} + \cdots + (r_{d'} - x^{(0)})\mathbb{R}\right\}$.

$\mathbb{R}$-*semi-linear.* Let us now generalise this approach to $\mathbb{R}$-semi-linear sets. The collection of $\mathbb{R}$-semi-linear sets, $\{\bigcup_{i=1}^{m} L_i \mid m \in \mathbb{N}, L_1, \ldots, L_m$ are $\mathbb{R}$-linear sets$\}$, 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 $\overline{X}^{\mathbb{R}}$ this smallest $\mathbb{R}$-semi-linear set. We are interested in $\overline{\mathcal{O}}^{\mathbb{R}}$.

Theorem 1. *The strongest $\mathbb{R}$-semi-linear invariant $\overline{\mathcal{O}}^{\mathbb{R}}$ of $\mathcal{O}$ is computable.*

Algebraic sets are those that are definable by finite unions and intersections of zeros of polynomials. For example, $\{(x, y) \mid xy = 0\}$ describes the lines $x = 0$ and $y = 0$. The (real) Zariski closure $\overline{X}^z$ of a set X is the smallest algebraic subset of $\mathbb{R}^d$ containing the set X. The Zariski closure of the set of reachable points, $\overline{\mathcal{O}}^z$, can be computed algorithmically [15].

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

[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 affects the original set.

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 (and in particular a Zariski closure) can be written effectively as a finite union of irreducible sets [3].

Proposition 1. *Let* $\overline{X}^z = A_1 \cup \cdots \cup A_k$, *with* A_i*'s irreducible. Then* $\overline{X}^{\mathbb{R}} = \overline{\overline{X}^z}^{\mathbb{R}} = \overline{A_1}^{\mathbb{R}} \cup \cdots \cup \overline{A_k}^{\mathbb{R}} = \overline{A_1}^a \cup \cdots \cup \overline{A_k}^a$.

Proof. Since $A_i \subseteq \overline{X}^{\mathbb{R}} = \cup_j L_j$, and A_i is irreducible, we have $A_i \subseteq L_j$ for some j (as the L_j's are algebraic sets). Since L_j is $\mathbb{R}$-linear, and $\overline{A_i}^a$ is the smallest $\mathbb{R}$-linear set covering A_i, we have $\overline{A_i}^a \subseteq L_j$. Taking $\overline{X}^{\mathbb{R}} = \overline{A_1}^a \cup \cdots \cup \overline{A_k}^a$ is thus optimal. □

Thus $\overline{\mathcal{O}}^{\mathbb{R}}$ can be obtained by computing $\overline{A_i}^a$ for each irreducible A_i, where $\overline{\mathcal{O}}^z = A_1 \cup \cdots \cup A_k$. To complete the proof of Theorem 1 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 let $W \leftarrow \overline{W \cup \{y\}}^a$, where $y \in A \setminus W$. 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 $\overline{A}^a = W$.

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

Recall that a $\mathbb{Z}$-linear set $\{q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\}$ 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., $\{p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\}$, in d-dimensional space, translated to start from q rather than $\mathbf{0}$.

Theorem 2. *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.*

The image of a $\mathbb{Z}$-linear set $L = \{q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\}$ by a matrix M is the $\mathbb{Z}$-linear set: $M(L) = \{Mq + (Mp_1)\mathbb{Z} + \cdots + (Mp_n)\mathbb{Z}\}$. The following lemma asserts that when two points are in a $\mathbb{Z}$-linear set, the direction between these two points can be applied from any reachable point, and hence this direction can be included as a period without altering the set.

Proposition 2. *Let* $L = \{q + a_1p_1 + \cdots + a_np_n \mid a_1, \ldots, a_n \in \mathbb{Z}\}$ *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* $L = \{q + a_1p_1 + \cdots + a_np_n + a'(y - x) \mid a_1, \ldots, a_n, a' \in \mathbb{Z}\}$.

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 $z + a'(y - x) \in L$. □

Proposition 3. *Given two $\mathbb{Z}$-linear sets $L_1 = \{q + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z}\}$ and $L_2 = \{s + t_1\mathbb{Z} + \cdots + t_m\mathbb{Z}\}$, there exists a smallest $\mathbb{Z}$-linear set L containing $L_1 \cup L_2$: the set $L = \{q + (s-q)\mathbb{Z} + p_1\mathbb{Z} + \cdots + p_n\mathbb{Z} + t_1\mathbb{Z} + \cdots + t_m\mathbb{Z}\}$.*

Proof. First we show $L_1 \cup L_2 \subseteq L$:

- If $x = q + a_1p_1 + \cdots + a_np_n \in L_1$, then $x = q + (s-q)0 + a_1p_1 + \cdots + a_np_n + 0t_1 + \cdots + 0t_m \in L$.
- If $x = s + b_1t_1 + \cdots + b_mt_m \in L_2$, then $x = q + (s-q)1 + 0p_1 + \cdots + 0p_n + b_1t_1 + \cdots + b_mt_m \in L$.

Next we show minimality as a straightforward consequence of Proposition 2.

Clearly the vectors $p_1, \ldots, p_n$ can be added by Proposition 2 because any two points of L_1 differing by p_i guarantees that adding p_i does not alter the resulting set. Similarly, $t_1, \ldots, t_m$ can also be included. Finally, by Proposition 2, the vector $s - q$ can be included because q and s both belong to $L_1 \cup L_2$. □

A d-dimensional lattice can always be defined by at most d vectors; and thus if d is the dimension of the matrices, no more than d period vectors are needed in total. However, Proposition 3 induces a representation which may over-specify the lattice by producing more than d vectors to define the lattice.

Example 2. 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 $L_i = \{p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\}$. 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[8].

These 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 form always exists, and the columns of H form a basis of the same lattice as 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 columns of a matrix in Hermite normal form constitute a unique basis for the lattice (up to additional redundant zero columns). Hence a basis of minimal dimension can be obtained by computing the Hermite normal form of the matrix formed by placing the period vectors into columns.

[8] 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.

We now prove the main theorem:

Proof (Proof of Theorem 2*).* We claim that Algorithm 1 returns the strongest $\mathbb{Z}$-linear invariant I.

Algorithm 1 proceeds in two phases:

- First find a necessary subset $L_0 \subseteq I$ of the invariant having already the same dimension as I.
- Then 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.

Recall the set $R_0 = \{x^{(0)}, r_1, \ldots, r_{d'}\} \subseteq \mathcal{O}$, with $d' \leq d$, from Lemma 1. The resulting $\mathbb{Z}$-linear set $L_0 = \{x^{(0)} + (r_1 - x^{(0)})\mathbb{Z} + \cdots + (r_{d'} - x^{(0)})\mathbb{Z}\}$ is then a d'-dimensional porous subset of the d'-dimensional affine hull of the orbit ($L_0 \subseteq \overline{\mathcal{O}}^a$). 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 2 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 $\mathcal{O}$. To do this we repeatedly apply the covering procedure of Proposition 3. That is, L_{i+1} is the smallest $\mathbb{Z}$-linear set covering $L_i \cup M_1(L_i) \cup \cdots \cup M_k(L_i)$. To keep the number of vectors small, we keep the period vectors of the $\mathbb{Z}$-linear set in Hermite normal form.

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 becoming fixed. The volume of the parallelepiped is bounded above by $|p_1| \cdots |p_{d'}|$.

At each step, the volume of the parallelepiped must at least halve, thus the volume at step t is $vol_t \leq |p_1| \cdots |p_{d'}|/2^t$. The procedure must saturate at or before the volume becomes 1, which occurs after at most $\log(|p_1| \cdots |p_{d'}|) = \sum_i \log(|p_i|)$ steps. At each step, for efficiency considerations, we convert the $\mathbb{Z}$-linear set into Hermite normal form to retain exactly d' period vectors.

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

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 vectors in L_0 can be present, without loss of generality, thanks to Proposition 2. Assume $L_i \subseteq J$. Then it must be the case that J contains every $M_j(L_i)$, 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 $M_j(L_i)$ for all $j \leq k$. Finally, since I is itself one of the L_i's, we have $I \subseteq J$ as required. □

Remark 1. Note that a $\mathbb{Z}$-linear set is not sufficient for the MU puzzle: both 1 and 2 are in the reachability set, thus $\{1 + 1\mathbb{Z}\} = \mathbb{Z}$ is the strongest $\mathbb{Z}$-linear invariant.

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

Input: $x^{(0)}, M_1, \ldots, M_k$
Compute $R_0 = \left\{x^{(0)}, r_1, \ldots, r_{d'}\right\} \subseteq \mathcal{O}$
Compute $p_i = r_i - x^{(0)}$ for $i \in \{1, \ldots, d'\}$
$L_0 = \left\{x^{(0)} + p_1\mathbb{Z} + \cdots + p_{d'}\mathbb{Z}\right\}$
while *True* **do**
  $L_i = \text{Covering}(L_{i-1} \cup M_1(L_{i-1}) \cup \cdots \cup M_k(L_{i-1}))$
  $H_i = \text{HermiteNormalForm}(L_i)$
  $L_i = \left\{x^{(0)} + h_1\mathbb{Z} + \cdots + h_{d'}\mathbb{Z} \mid h_j \text{ column of } H_i\right\}$
  if $L_i = L_{i-1}$ **then**
    return L_i
  end
end

4.1 Extensions of $\mathbb{Z}$-linear Sets Without Strongest Invariants

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

$\mathbb{Z}$-semi-linear sets are unions of $\mathbb{Z}$-linear sets, and therefore can include singletons. 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 $\mathcal{O} = \left\{2^k \mid k \in \mathbb{N}\right\}$, which is not even $\mathbb{N}$-semi-linear (our most general class). 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 an improvement.

$\mathbb{N}$-linear sets generalise $\mathbb{Z}$-linear sets (observe that $\mathbb{Z}$-linear sets are a proper subclass, since $\{x + p_i\mathbb{Z}\}$ can be expressed as $\{x + (-p_i)\mathbb{N} + p_i\mathbb{N}\}$, but $\{x + p_i\mathbb{N}\}$ is clearly not $\mathbb{Z}$-linear). Consider the LDS $((x_1, x_2), \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right))$, 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 $\{y + (x - y)\mathbb{R}_+\}$ and $\{x + (y - x)\mathbb{R}_+\}$ are incomparable half-lines.

4.2 $\mathbb{Z}$-linear Targets

We have so far only considered invariants for point targets. We now turn to lattice-like targets, in particular targets specified as *full-dimensional* $\mathbb{Z}$-linear sets.

Theorem 3. *It is decidable whether a given LDS $(x^{(0)}, \{M_1, \ldots, M_k\})$ reaches a full-dimensional $\mathbb{Z}$-linear target $Y = \{x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\}$, with $x, p_i \in \mathbb{Z}^d$.*

Furthermore, for unreachable instances, a $\mathbb{Z}$-semi-linear inductive invariant can be provided.

Theorem 3 requires the targets to be *full-dimensional*. For nondeterministic systems reachability is undecidable for non-full-dimensional targets (in particular point targets) [22]. However, even for deterministic systems, when $\mathbb{Z}$-linear targets fail to be *full-dimensional* the reachability problem becomes as hard as the Skolem problem (see, e.g. [24]), for example by choosing as target the set $\{(0, x_2, \ldots, x_d) \mid x_2, \ldots, x_d \in \mathbb{Z}\} = \{\mathbf{0} + e_2\mathbb{Z} + \cdots + e_d\mathbb{Z}\}$, where $e_i \in \{0,1\}^d$ is the standard basis vector, with $(e_i)_i = 1$ and $(e_i)_j = 0$ for $i \neq j$.

Towards proving Theorem 3, 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). By square, we mean that all period vectors are the same multiple of standard basis vectors.

Lemma 2. *Let $Y = \{x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z}\}$ be a full-dimensional $\mathbb{Z}$-linear set. Then there exists $m \in \mathbb{N}$ and a finite set $B \subseteq [0, m-1]^d$ such that $Y = \bigcup_{b\in B} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$.*

Proof. Suppose $p_1, \ldots, p_d$ span a d-dimensional vector space. Let $P = \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix}$ be the matrix with rows $p_1, \ldots, p_d$. Since P is full row rank it is invertible, hence there exists a rational matrix P^{-1} such that $e_i = P^{-1}_{i,1}p_1 + \cdots + P^{-1}_{i,d}p_d$. 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 2, Y is equivalent to $\{x + p_1\mathbb{Z} + \cdots + p_d\mathbb{Z} + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$. 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 $x'_i = (x_i \mod m)$.

And therefore Y can be written as $\bigcup_{b\in B} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$, where $B = [0, m-1]^d \cap Y$. □

We now prove Theorem 3.

Proof (Proof of Theorem 3). Choose m and B as in Lemma 2, so that Y is of the form $\bigcup_{b\in B} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$. We build an invariant I of the form $\bigcup_{b\in B'} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$ for some $B' \subseteq [0, m-1]^d$.

We initialise the set $I_0 = \{x + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$, where $x \in [0, m-1]^d$ such that $x_j = (x^{(0)}_j \mod m)$. We then build the set I_1 by adding to I_0 the sets $\{y + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$ where for each choice of M_i, $y \in [0, m-1]^d$ is formed by $y_j = ((M_i x)_j \mod m)$ 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 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, wind up inside the set $\{y + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$, which is a part of the target.

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

Remark 2. By the same argument, Theorem 3 extends to a restricted class of $\mathbb{Z}$-semi-linear targets: the finite union of *full-dimensional* $\mathbb{Z}$-linear sets.

5 $\mathbb{N}$-Semi-linear Invariants

We now consider $\mathbb{N}$-semi-linear invariants, our most general class. $\mathbb{N}$-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, $\mathbb{N}$, is captured exactly by an $\mathbb{N}$-linear invariant. We show that a separating $\mathbb{N}$-semi-linear invariant can *always* be found for unreachable instances of deterministic integer LDS, although the computed invariant will depend on the target. However, finding invariants is undecidable for nondeterministic systems, at least in high dimension. Nevertheless, we show 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 [16]. In this subsection, we establish the following result to provide a separating invariant in unreachability instances.

Theorem 4. *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.*

To do so, we will invoke the results from [8] 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 [8] 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 3. *Given an $\mathbb{R}_+$-linear set $S = \{x + \sum_i p_i\mathbb{R}_+\}$, 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 (Proof of Theorem 4). We note that every invariant produced in [8] has rational period vectors, as the vectors are given by the difference of successive point in the orbit of the system, and thus Lemma 3 can be applied. The authors of [8] 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 modulus 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 [19] asserts that unless all of the eigenvalues are roots of unity, one of them must have modulus strictly greater than 1 (the case in which *all* eigenvalues are 0 being of course trivial).

This concludes the proof of Theorem 4. □

5.2 Undecidability of $\mathbb{N}$-semi-linear Invariants for Nondeterministic LDS

If the enhanced expressivity of $\mathbb{N}$-semi-linear sets allows us always to 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) that can be defined in the following way: given m pairs of non-empty words $\{(u^1, v^1), \ldots, (u^m, v^m)\}$ over alphabet $\{0, 2\}$, does there exist an infinite word $w = w_1 w_2 \ldots$ over alphabet $\{1, \ldots, m\}$ such that $u^{w_1} u^{w_2} \ldots = v^{w_1} v^{w_2} \ldots$. This problem is known to be undecidable when m is at least 8 [6,13].

Theorem 5. *The invariant synthesis problem for $\mathbb{N}$-semi-linear sets and linear dynamical systems with at least two matrices of size* 91 *is undecidable.*

Proof (Sketch). We first establish the result in the case of several matrices in low dimension; this can then be transformed in a standard way to two larger matrices (of size 91).

The proof is by reduction from the infinite Post correspondence problem. Given an instance of this problem the pair of words corresponding to each sequence of tiles has an integer representation, using base-4 encoding. An important property of our encoding is that the operation of appending a new tile to an existing pair of words can be encoded 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 encoding is such that this difference of letters creates a difference in their numerical encodings that can be identified with an $\mathbb{N}$-semi-linear invariant. On the other hand, when there is a positive answer to the ω-PCP instance, there can be no $\mathbb{N}$-semi-linear invariant. □

5.3 Nondeterministic One-Dimensional Affine Updates

The previous section shows that point reachability for nondeterministic LDS is undecidable once there 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 6. *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.

We note that decidability of reachability is already known [9,10]. We refine this result by exhibiting an invariant which can be used to disprove 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. 6.

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

Definition 5. *A function $f(x) = ax + b$...*

- *... is* redundant *if $f(x) = b$, (including possibly $b = 0$), or if $f(x) = x$.*
- *... is* counter-like *if $f(x) = x + b$, $b \neq 0$. Two counter-like functions, $f(x) = x + b$ and $g(x) = x + c$ are* opposing *if $b > 0$ and $c < 0$ (or vice-versa).*
- *... is* growing *if $f(x) = ax + b$ and $|a| \geq 2$. We say a growing function is* inverting *if $a \leq -2$.*
- *... is* pure inverting *if $f(x) = -x + b$.*

Simplifying Assumptions

Lemma 4. *Without loss of generality, redundant functions are redundant; more precisely, we can reduce the computation of an invariant for a system having redundant functions to finitely many invariant computations for systems devoid of 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 5. *Without loss of generality, $x^{(0)} \geq 0$.*

Proof. We construct a new system, where each transition $f(x) = ax + b$ is replaced by $\overline{f}(x) = ax - b$. 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 $\overline{f}(-x) = -ax - b = -f(x)$. □

Lemma 6. *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. □

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

Lemma 7. *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 $d = \gcd(b, c)$.*

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

Lemma 8. *Let $h(x) = x + d$ be chosen as a reference counter amongst the counters. If $\{x + d\mathbb{Z}\} \in I$, then $\{f(x) + d\mathbb{Z}\} \in I$ for every function f.*

Proof (Proof of Lemma 8*).* Consider the function $f(x) = ax + b$. If $x = y + dk \in I$, then $f(x) = ax + b = ay + adk + b = f(y) + adk \in I$.

Now thanks to the presence of counter $h(x) = x + d$, by choosing the initial $k \in \mathbb{Z}$ appropriately and applying $h(x)$ sufficiently many times (say $m \in \mathbb{N}$ times), one can reach $f(x) + adk + dm = f(x) + dn$ for any desired $n \in \mathbb{Z}$. □

Without loss of generality if $\{x + d\mathbb{Z}\}$ is in the invariant, then $0 \leq x < d$. We then repeatedly use Lemma 8 to find the required elements of the invariant. Since there are only finitely many residue classes (modulo d), every reachable residue class $\{c_1, \ldots, c_n\}$ can be found by saturation (in at most d steps), yielding invariant $\{c_1 + d\mathbb{Z}\} \cup \cdots \cup \{c_n + d\mathbb{Z}\}$.

Thanks to Lemma 6, in all remaining cases there is without loss of generality at most one pure inverter.

Only Pure Inverters. If there is exactly one pure inverter $f(x) = -x + b$ (and no other types of functions), then $f(x^{(0)}) = -x^{(0)} + b$ and $f(-x^{(0)} + b) = x^{(0)} - b + b = x^{(0)}$, thus the reachability set is finite, with exact invariant $\left\{x^{(0)}, -x^{(0)} + b\right\}$.

No Counters. If we are not in the preceding case and there are no counters, then there must be growing functions and by Lemma 6, without loss of generality at most one pure inverter. We show that all growing functions increase the modulus outside of some bounded region.

Lemma 9. *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 $C = \max\left\{C^0_{f_1}, \ldots, C^0_{f_k}, |y| + 1\right\}$, for $f_1, \ldots, f_k$ growing functions. If there are no pure inverters then $\{-C - \mathbb{N}\} \cup \{C + \mathbb{N}\}$ is invariant (although may not yet contain the whole of $\mathcal{O}$). However, we can return 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 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 $C' = \max\left\{C^{|d|}_{f_1}, \ldots, C^{|d|}_{f_k}, |y| + 1\right\} + |d|$. Then the invariant is $\{-C' - \mathbb{N}\} \cup \{C' + d + \mathbb{N}\} \cup (\mathcal{O} \cap (-C', C' + d))$.

Non-opposing 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-like functions, but they are all counting in the same direction. There may also be a single pure inverter, and possibly some growing functions.

Pick a counter $h(x) = x+d$ to be the reference counter; the choice is arbitrary, but it is convenient to pick a counter with minimal $|d|$. As a starting point, we have $\left\{x^{(0)} + d\mathbb{N}\right\} \subseteq I$.

Lemma 10. *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 $\{g(x) + d\mathbb{Z}\} \subseteq I$.*

The crucial difference with Lemma 8 is the observation that now an $\mathbb{N}$-linear set has induced a $\mathbb{Z}$-linear set.

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 $h(x)$. Hence for any $m \in \mathbb{Z}$ there exists $k, n \in \mathbb{N}$ such that $k - na = m$, so that r is indeed reachable. □

Similarly to the situation with two opposing counters, whenever the invariant contains some $\mathbb{Z}$-linear set, Lemma 8 allows us to saturate amongst the finitely many reachable residue classes.

However, the invariant may contain subsets that are not $\mathbb{Z}$-linear. Consider $\{x + d\mathbb{N}\} \subseteq I$, which is not yet invariant. We repeatedly apply non-inverting functions to $\{x + d\mathbb{N}\}$ to obtain new $\mathbb{N}$-linear sets (not $\mathbb{Z}$-linear sets). When the function applied 'moves' in the direction of the counters this will ultimately saturate (in particular when applying other counter functions). However, in the opposite direction, we may generate infinitely many such classes.

Example 3. 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 $\{2 + 4\mathbb{Z}\}$. □

Clearly we can examine all reachable residue classes defined by our reference counter. Any residue class reachable after an inverting function induces a $\mathbb{Z}$-linear set. So it remains to consider those $\mathbb{N}$-linear sets reachable without inverting functions. The remaining case to handle occurs when we repeatedly induce $\mathbb{N}$-linear sets until they repeat a residue class in the direction opposite to that of the reference counter.

We consider the case for $h(x) = x + d$ with $d \geq 0$. The case with $h(x) = x - d$ is symmetric. It remains to detect when a set $\{x + d\mathbb{N}\}$ leads to $\{y + d\mathbb{N}\}$ by a sequence of non-inverting functions with $x \equiv y \mod d$. Then by repeated application of these functions one can reach sets $\{z + d\mathbb{N}\}$ with z arbitrarily small, hence we can replace $\{x + d\mathbb{N}\}$ by $\{x + d\mathbb{Z}\}$. We give further details in the full version.

Reachability. The above procedure is sufficient to decide reachability. In all cases apart from that 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 case, 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 modulus, a different invariant would need to be built.

Complexity

Lemma 11. *Assume that all functions, starting point, and target point are given in unary. Then the invariant can be computed in polynomial time.*

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, for example, $(-C, C)$, where C is of size polynomial in the largest value.

As shown in [10], 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 The POROUS Tool

Our invariant-synthesis tool POROUS[9] 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 3 (restricted to one-dimensional affine systems) and Theorem 6. POROUS is built in Python and can be used by command-line file input, a web interface, or by directly invoking the Python packages.

POROUS takes as input an instance (a start 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).

[9] Tool: invariants.davidpurser.net Code: github.com/davidjpurser/porous-tool.

Table 2. Results varying by size parameter (last row includes all instances tested). Times are given in seconds, with the average and maximum shown (except reachability proof time, which are all approximately 30 s due to instances that terminate just before the timeout).

Size	Invariant build time		Unreachable instances	Invariant proof time		Reachable instances	Reachable with proofs	Reachability proof time
	Avg	Max		Avg	Max		Within ≈30s	Avg
8	0.001	0.009	100 (9.84%)	0.005	0.261	916 (90.2%)	911 (99.5%)	0.033
16	0.001	0.020	122 (12.0%)	0.010	0.788	894 (88.0%)	885 (99.0%)	0.053
32	0.003	0.068	134 (13.2%)	0.020	0.911	882 (86.8%)	843 (95.6%)	0.203
64	0.008	0.261	150 (14.8%)	0.052	2.969	866 (85.2%)	766 (88.5%)	0.294
128	0.021	0.557	153 (15.1%)	0.096	2.426	863 (84.9%)	719 (83.3%)	0.464
256	0.088	2.838	166 (16.3%)	0.316	43.587	850 (83.7%)	620 (72.9%)	0.998
512	0.428	9.312	162 (15.9%)	0.899	21.127	854 (84.1%)	570 (66.7%)	1.120
1024	1.121	20.252	173 (17.0%)	3.275	65.397	843 (83.0%)	514 (61.0%)	1.646
All	0.209	20.252	1160 (14.3%)	0.584	65.397	6968 (85.7%)	5828 (83.6%)	0.499

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 modulus of a and b, ranging between 8 and 1024. The starting point was between 1 and the size parameter and the target was between 1 and 4 times the size parameter. Ten 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).

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, 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 30 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 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 84% produced a proof within 30 s). Of the 14% of unreachable instances, all produced a proof, with the invariant taking around 0.2 s to build and 0.6 s to produce the proof. The 30-s timeout when demonstrating reachability directly is several orders of magnitudes longer than answering the reachability query via our invariant-building method.

A typical academic/consumer laptop was used to conduct the timing and analysis (a four-year-old, four-core MacBook Pro).

7 Conclusions and Open Directions

We 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 in particular focused on reachability questions. Clearly, 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 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 Fellowship EP/N008197/1.

References

1. Almagor, S., Chistikov, D., Ouaknine, J., Worrell, J.: O-minimal invariants for discrete-time dynamical systems (2019, preprint, submitted). https://arxiv.org/abs/1802.09263

2. Bozga, M., Iosif, R., Konecný, F.: Fast acceleration of ultimately periodic relations. In: Touili, T., Cook, B., Jackson, P.B. (eds.) Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, 15–19 July 2010. Proceedings. Lecture Notes in Computer Science, vol. 6174, pp. 227–242. Springer, Heidelberg (2010). 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
3. Chistov, A.: Algorithm of polynomial complexity for factoring polynomials and finding the components of varieties in subexponential time. J. Soviet Math. **34**(4), 1838–1882 (1986). https://doi.org/10.1007/BF01095643
4. Clarke, E.M., et al.: Abstraction and counterexample-guided refinement in model checking of hybrid systems. Int. J. Found. Comput. Sci. **14**(4), 583–604 (2003). https://doi.org/10.1142/S012905410300190X
5. Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: Aho, A.V., Zilles, S.N., Szymanski, T.G. (eds.) Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978, pp. 84–96. ACM Press (1978). https://doi.org/10.1145/512760.512770
6. Dong, J., Liu, Q.: Undecidability of infinite post correspondence problem for instances of size 8. RAIRO Theor. Informatics Appl. **46**(3), 451–457 (2012). https://doi.org/10.1051/ita/2012015
7. Douglas, R.H.: Gödel, Escher, Bach: An Eternal Golden Braid. Basic Books, New York (1979)
8. Fijalkow, N., Lefaucheux, E., Ohlmann, P., Ouaknine, J., Pouly, A., Worrell, J.: On the Monniaux problem in abstract interpretation. In: Chang, B.-Y.E. (ed.) SAS 2019. LNCS, vol. 11822, pp. 162–180. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32304-2_9
9. Finkel, A., Göller, S., Haase, C.: Reachability in register machines with polynomial updates. In: Chatterjee, K., Sgall, J. (eds.) MFCS 2013. LNCS, vol. 8087, pp. 409–420. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40313-2_37
10. Fremont, D.: The reachability problem for affine functions on the integers. CoRR abs/1304.2639 (2013). http://arxiv.org/abs/1304.2639
11. Giesl, J., et al.: Analyzing program termination and complexity automatically with AProVE. J. Autom. Reason. **58**(1), 3–31 (2016). https://doi.org/10.1007/s10817-016-9388-y
12. Ginsburg, S., Spanier, E.H.: Bounded ALGOL-like languages. Trans. Am. Math. Soc. **113**(2), 333–368 (1964). https://doi.org/10.1090/S0002-9947-1964-0181500-1
13. Halava, V., Harju, T.: Undecidability of infinite post correspondence problem for instances of size 9. RAIRO Theor. Informatics Appl. **40**(4), 551–557 (2006). https://doi.org/10.1051/ita:2006039
14. Heizmann, M., Hoenicke, J., Podelski, A.: Termination analysis by learning terminating programs. In: Biere, A., Bloem, R. (eds.) CAV 2014. LNCS, vol. 8559, pp. 797–813. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08867-9_53
15. Hrushovski, E., Ouaknine, J., Pouly, A., Worrell, J.: 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, 09–12 July 2018, pp. 530–539. ACM (2018). https://doi.org/10.1145/3209108.3209142
16. Kannan, R., Lipton, R.J.: Polynomial-time algorithm for the orbit problem. J. ACM **33**(4), 808–821 (1986). https://doi.org/10.1145/6490.6496

17. Karr, M.: Affine relationships among variables of a program. Acta Informatica **6**, 133–151 (1976). https://doi.org/10.1007/BF00268497
18. Kincaid, Z., Breck, J., Cyphert, J., Reps, T.W.: Closed forms for numerical loops. Proc. ACM Program. Lang. **3**(POPL), 55:1–55:29 (2019). https://doi.org/10.1145/3290368
19. Kronecker, L.: Zwei Sätze über Gleichungen mit ganzzahligen Coefficienten. Journal für die reine und angewandte Mathematik **57**(53), 173–175 (1857)
20. Leroux, J.: The general vector addition system reachability problem by presburger inductive invariants. Log. Methods Comput. Sci. 6(3) (2010). https://doi.org/10.2168/LMCS-6(3:22)2010
21. Leroux, J.: 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, 26–28 January 2011, pp. 307–316. ACM (2011). https://doi.org/10.1145/1926385.1926421
22. Markov, A.: On certain insoluble problems concerning matrices. Doklady Akad. Nauk SSSR. **57**, 539–542 (1947)
23. Monniaux, D.: On the decidability of the existence of polyhedral invariants in transition systems. Acta Informatica **56**(4), 385–389 (2018). https://doi.org/10.1007/s00236-018-0324-y
24. Ouaknine, J., Worrell, J.: Decision problems for linear recurrence sequences. In: Finkel, A., Leroux, J., Potapov, I. (eds.) RP 2012. LNCS, vol. 7550, pp. 21–28. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33512-9_3
25. Shmonin, G.: Lattices and Hermite normal form, February 2009. Lecture notes for the course Integer Points in Polyhedra at the Swiss Federal Institute of Technology Lausanne (EPFL)
26. Tzeng, W.: A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM J. Comput. **21**(2), 216–227 (1992). https://doi.org/10.1137/0221017