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, Germany
[2] Department of Computer Science, Oxford University, 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 · Invariants · Reachability · Presburger.

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*:

Meta Problem 1. *Consider a system with update functions $f_1, \dots, f_n$. 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 an inductive invariant I such that $x \in I$ and $Y \cap I = \emptyset$.*

We note that the existence of *smallest* inductive invariants is a desirable property; unfortunately, these are not always guaranteed to exist.

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. In prior work from the literature, typical classes of invariants are usually convex, or finite unions of convex sets. In this paper we consider 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[3]. We shall work instead with the equivalent formulation of semi-linear sets, generalising ultimately periodic sets to higher dimensions, as finite unions of sets of the form $\{b + p_1\mathbb{N} + \cdots + p_m\mathbb{N}\}$ (see Definition 2).

Let us first consider a motivating example:

Example 1 (Hofstadter's MU Puzzle [7]). Consider the following term-rewriting puzzle. Start with the word MI, and by applying the following grammar rules, we ask whether the word MU can ever be reached.

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

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 $\{x : x \cong 0 \mod 3\}$. 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,[4] or a two-dimensional system with two linear updates.[5] 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 considered as an example algorithmic verification problem in [4]; the tools considered there rely upon the manual provision of an

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

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

[5] $\begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} ax + b \\ 1 \end{pmatrix}$ models affine functions using a matrix representation, but one of the dimensions is always fixed to 1.

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 $x \mapsto 2x \mid\mid x \mapsto x - 3$).

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

- We first consider 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 linear and semi-linear invariants for point targets is given in Table 1. For completeness we also consider the $\mathbb{R}, \mathbb{R}_+$-(semi)-linear sets, where we complete the picture from prior work by showing that strongest $\mathbb{R}$-semi-linear invariants are computable.
    - We show the existence of *strongest* $\mathbb{Z}$-linear invariants, and show that they can be found in polynomial time. 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 a $\mathbb{N}$-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 for any point target for *deterministic* integer linear dynamical systems. In comparison with the $\mathbb{Z}$-linear case, this sacrifices speed of computation and simplicity of representation for accuracy.
    - However, for nondeterministic integer linear dynamical systems, computing a $\mathbb{N}$-semi-linear invariants is an undecidable problem in arbitrary dimension. Nevertheless we show how such invariants can be constructed in a low-dimension setting, and in particular for affine updates in one dimension. 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* $\mathbb{Z}$-linear targets we show in Theorem 3 that reachability is decidable, and, in the case of unreachability a $\mathbb{Z}$-semi-linear invariant can always be exhibited as a certificate. If the target is not *full-dimensional* then the reachability problem is Skolem-hard and undecidable for deterministic and nondeterministic systems respectively.
- In Section 6 we present a tool which handles the 1-d affine case 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 for a single affine update, or deterministic linear dynamical system, (in arbitrary dimension) is decidable in polynomial time for a point target (that is $Y = \{y\}$), as shown by Kannan and Lipton [16]. However,

Ring	D/N	**Linear**	**Semi-linear** (SL)
$\mathbb{Z}$	det	Strongest computable in P-time (Thm. 2).	No strongest (Sec. 4.1). Subsumed by $\mathbb{N}$-SL.
$\mathbb{Z}$	non	Strongest computable in P-time (Thm. 2).	No strongest (Sec. 4.1).
$\mathbb{N}$	det	No strongest (Sec. 4.1). Subsumed by $\mathbb{N}$-SL.	No strongest, but sufficient computable (Thm. 4).
$\mathbb{N}$	non	No strongest (Sec. 4.1).	1d-affine decidable (Thm. 6). Undecidable in general (Thm. 5).
$\mathbb{R}$	det	Strongest: Affine relations by Karr [17].	Strongest: Affine Closure on Zariski Closure (Thm. 1).
$\mathbb{R}$	non	Strongest: Affine relations by Karr [17].	Strongest: Affine Closure on Zariski Closure (Thm. 1).
$\mathbb{R}_+$	det	No strongest (Sec. 4.1). Subsumed by $\mathbb{R}_+$-SL.	No strongest, but sufficient computable [8].
$\mathbb{R}_+$	non	No strongest (Sec. 4.1).	Undecidable [8]

Table 1. Results for integer dynamical systems for a point target. Det/Non refers to deterministic or nondeterministic LDS. By ‘Subsumed by ...’ we mean there is a result which will generate a sufficient invariant, although it will be of a more general type.

the nondeterministic variant is undecidable in general, where the choice of M at each step is chosen nondeterministically at each time step from a finite set. In the nondeterministic case the reachability problem is strongly related to the matrix semi-group membership problem, which is undecidable [20], and so increasing the dimension quickly leads to undecidability for reachability.

In particular this means that in the nondeterministic setting we cannot hope *always* to compute a separating invariant (as this would answer the reachability question). 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 where reachability is decidable (for example low dimension). 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 [21,8], algebraic [15] and o-minimal [1] invariants. For some types of invariants (e.g. algebraic [15]), it can still be decidable whether there is an invariant for unreachable instances, despite the reachability problem being undecidable (that is, there may be unreachable instances which admit an algebraic invariant, in which case the invariant can be found, or the instance may be unreachable but without admitting any algebraic invariant). Other works (e.g., [5]) use heuristics approaches to generate invariants for some instances, but we focus on the problem of generating invariants for all instances.

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 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.

2 Preliminaries

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

Definition 1 (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 $M_1, \dots, M_k \subseteq \mathbb{Z}^{d \times d}$. An LDS is* deterministic *if it has a single matrix ($k = 1$) and otherwise* nondeterministic.

A point y is reachable *if there exists $m \in \mathbb{N}$ and $B_1, \dots, B_m$ s.t. $B_1 \times \dots \times B_m x^{(0)} = y$ and $B_i \in \{M_1, \dots, M_k\}$ 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_i is defined by a base vector $b \in \mathbb{Z}^d$ and period vectors $p_1, \dots, p_d \in \mathbb{Z}^d$ such that*

$$L_i = \{b + a_1 p_1 + \dots + a_d p_d \mid a_1, \dots, a_d \in \mathbb{K}\} .$$

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

$\mathbb{N}$-semi-linear sets correspond to Presburger arithmetic [12]. However, we can also consider $\mathbb{Z}$ (the variant without order), and the real variants ($\mathbb{R}$ 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 $(x^{(0)}, \{M_1, \dots, M_k\})$ then a set I is an inductive invariant if*

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

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

Definition 4 (Invariant Synthesis Problem). *Given an invariant domain $\mathcal{D}$, an integer linear dynamical system $(x^{(0)}, \{M_1, \dots, M_k\})$ and a target Y, does there exist an inductive invariant I in $\mathcal{D}$, disjoint from Y?*

In our setting, we are interested in $\mathcal{D}$ that are linear, and semi-linear.

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

Before delving into porous invariants, let us consider invariants in the real variants of our settings. That is, we consider invariants described as $\mathbb{R}$-(semi)-linear sets—these invariants are not porous but have a similar description language.

In fact, $\mathbb{R}$-linear invariants are equivalent to the affine hull of the points, as found by Karr's algorithm [17]. In the first subsection we recall a key result—in later subsections this will form a starting point for our $\mathbb{Z}$-linear porous invariants. Secondly, we complete the picture for real invariants, by showing $\mathbb{R}$-semi-linear invariants can be computed by combining techniques for algebraic invariants [15] and the $\mathbb{R}$-linear invariants we establish first.

$\mathbb{R}$-*linear* Recall, a set is $\mathbb{R}$-linear if $L = \{v_{i,0} + v_{i,1}\mathbb{R} + \cdots + v_{i,t}\mathbb{R}\}$ for some $v_{i,0}, \ldots, v_{i,t} \in \mathbb{Z}^d$ that can be assumed linearly-independent[6] without loss of generality (and so $t \leq d$). Given two points of L, every point on the line formed by them must 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 recall the reachable set of vectors, $\mathcal{O} = \left\{ M_{i_m} \cdots M_{i_1} x^{(0)} \mid m \in \mathbb{N}, i_1, \ldots, i_m \in \{1, \ldots, k\} \right\}$. Computing $\overline{\mathcal{O}}^a$ is exactly the smallest $\mathbb{R}$-linear invariant under $\mathbb{R}$. Karr's algorithm [17] can be used to find this smallest invariant, in polynomial time [24]. The following follows from Theorem 3.1 of [24].

Lemma 1 (proof in Appendix A). *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 of the absolute values of the integers appearing in $x^{(0)}$ and $M_1, \ldots, M_k$), a $\mathbb{Q}$-affinely independent set of 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 $\mu_0 := (d\mu)^d$.*

Assume the resulting computation from Lemma 1 is $R_0 = \left\{ x^{(0)}, r_1, \ldots, r_{d'} \right\}$, with $d' \leq d$. The $\mathbb{R}$-linear invariant 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 to $\mathbb{R}$-semi-linear sets. The set of $\mathbb{R}$-semi-linear sets, $\{\bigcup_{i=1}^{m} L_i \mid m \in \mathbb{N}, \quad L_1 \ldots L_m \text{ are } \mathbb{R}\text{-linear sets}\}$, is closed under finite union and arbitrary intersection[7]. This implies for any candidate collection

[6] $v_1, \ldots, v_m$ are linearly independent if there does not exist $a_1, \ldots, a_m \in \mathbb{R}$, not all 0, s.t. $a_1 v_1 + \cdots + a_m v_m = 0$.

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

of $\mathbb{R}$-semi-linear covering a set X, there is a smallest $\mathbb{R}$-semi-linear covering X, namely the intersection. We denote by $\overline{X}^{\mathbb{R}}$ this smallest $\mathbb{R}$-semi-linear set such that $X \subseteq \overline{X}^{\mathbb{R}}$. We are interested in $\overline{\mathcal{O}}^{\mathbb{R}}$.

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

Algebraic sets are the sets 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 Zariski closure $\overline{X}^{z}$ of a set X is the smallest algebraic set covering the set X. The Zariski closure of the set of reachable point, $\overline{\mathcal{O}}^{z}$, can be computed [15].

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 $A \subseteq C$. Given an algebraic set, in particular a Zariski closure $\overline{X}^{z}$, can be written effectively as a union of irreducible sets $A_1, \ldots, A_k$ such that $\overline{X}^{z} = A_1 \cup \cdots \cup A_k$ [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 (in particular 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 $\overline{A_i}^{a} \subseteq L_i$. Taking $\overline{X}^{\mathbb{R}} = \overline{A_1}^{a} \cup \cdots \cup \overline{A_k}^{a}$ is thus optimal. □

Thus $\overline{\mathcal{O}}^{\mathbb{R}}$ is computable by computing $\overline{A_i}^{a}$ for each irreducible A_i, where $\overline{\mathcal{O}}^{z} = A_1 \cup \cdots \cup A_k$. We confirm in Appendix B that Karr's algorithm can be applied to algebraic sets.

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

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

Theorem 2. *Given a dynamical system $(x^{(0)}, \{M_1, \ldots, M_k\})$ we show there exists a smallest $\mathbb{Z}$-linear inductive invariant for the set of reachable points $\mathcal{O}$.*

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: $M(L) = \{Mq + (Mp_1)\mathbb{Z} + \ldots (Mp_n)\mathbb{Z}\}$. The following lemma shows 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 so this direction can be included as a period without expanding the set.

Proposition 2. *Given a $\mathbb{Z}$-linear set L. 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, the following sets are equivalent*

- $L = \{q + a_1p_1 + \cdots + a_np_n \mid a_1, \ldots, a_n \in \mathbb{Z}\}$

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

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 $z = q + c_1p_1 + \cdots + c_np_n$, then also $z + a'(y-x) = q + c_1p_1 + \cdots + c_np_n + a'((b_1 - a_1)p_1 + \cdots + (b_n - a_n)p_n) = q + (c_1 + a'(b_1 - a_1))p_1 + \cdots + (c_n + a'(b_n - a_n))p_n)$ 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 $L_1 = \{q + p_1\mathbb{Z} + \ldots p_n\mathbb{Z}\}$ and $L_2 = \{s + t_1\mathbb{Z} + \ldots t_m\mathbb{Z}\}$ there exists a smallest $\mathbb{Z}$-linear set L covering $L_1 \cup L_2$: the set $L = \{q + (s-q)\mathbb{Z} + p_1\mathbb{Z} + \ldots p_n\mathbb{Z} + t_1\mathbb{Z} + \ldots t_m\mathbb{Z}\}$.*

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 $x = sb_1t_1 + \cdots + b_mt_m \in L_2$, then $x = q + (s-q)1 + 0p_1 + \cdots + 0p_n + b_1t_1 + \cdots + b_mt_m \in L$

Next we show this is minimal which is a quick 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 can be used to argue p_i can be included. Similarly, $t_1, \ldots t_m$ can stay by Proposition 2. Finally, by Proposition 2, the vector $s - q$ can be included because q and s are inside $L_1 \cup L_2$. □

A d-dimensional lattice can always be defined by at most d vectors; and so if d is the dimension of the matrices, we do not need more than d period vectors 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, it is equivalent to the lattice $\{(2,0)\mathbb{Z} + (0,2)\mathbb{Z}\}$. Note that this is not equivalent to choosing an independent subset of the periods, as none of the following are equivalent: $\{(2,2)\mathbb{Z} + (0,6)\mathbb{Z}\}$, $\{(2,2)\mathbb{Z} + (2,6)\mathbb{Z}\}$, $\{(0,6)\mathbb{Z} + (2,6)\mathbb{Z}\}$ 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 is equivalent if p_i is swapped with p_j, if p_i is replaced by $-p_i$, and if p_i is replaced by $p_i + \alpha p_j$ where α is any fixed integer[8].

These are the uni-modular operations. The Hermite normal form of a matrix M is H such that $M = UHU^{-1}$, 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

[8] the last is the trickiest to see, but $x = y + \beta p_i \in L$ then $x = y + \beta(p_i + \alpha p_j) - \beta\alpha p_j$ is in the new lattice.

columns of M, because they differ up to unimodular (lattice preserving) operations. There are many text on the subject, but the lecture notes of Shmonin [23] give a particularly clear explanation.

If a matrix is in Hermite normal form then the columns of the matrix form a unique basis for the lattice, (up to additional redundant zero columns). Therefore a minimal dimensional basis can be found by finding the Hermite normal form of the matrix formed by placing the period vectors into columns.

We now prove the main theorem:

Proof (Proof of Theorem 2). Algorithm 1 shows the procedure. We prove the theorem with two claims: computation and termination in polynomial time and correctness.

Claim. The smallest $\mathbb{Z}$-linear invariant I can be computed in time polynomial in d, k and $\log \mu$.

The procedure proceeds in two phases:

- First find a necessary subset of the invariant $L_0 \subseteq I$ whose dimension already equals I.
- We then apply a chain of steps to find I such that $L_0 \subsetneq L_1 \subsetneq \cdots \subsetneq L_m = I$. Each step will only change the density to 'fill in' the required point in the invariant, and the number of steps m will be polynomial.

Recall the set R_0 from Lemma 1, let $R_0 = \left\{x^{(0)}, r_1, \ldots, r_{d'}\right\}$, with $d' \leq d$. From it, we build the $\mathbb{Z}$-linear set $L_0 = \left\{x^{(0)} + (r_1 - x^{(0)})\mathbb{Z} + \cdots + (r_{d'} - x^{(0)})\mathbb{Z}\right\}$. This is a d'-dimensional porous subset of the d'-dimensional affine hull of the orbit ($L_0 \subseteq \overline{\mathcal{O}}^a$), and hence remains d' dimension, and the dimension does not change under $M_1, \ldots, M_k$ (but the density may change). As each r_i and $x^{(0)}$ are in $\mathcal{O}$, by Proposition 2 we can assume each of the directions $(r_1 - x^{(0)})$ must be represented in the $\mathbb{Z}$-linear set, and therefore $L_0 \subseteq I$ for any inductive invariant I.

In step 2. 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 hermit 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 finite number of 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 $|p_1| \times |p_2| \times \cdots \times |p_{d'}|$. Each p_i was computed in polynomial time and has size polynomial in $\log((d\mu)^d)$, thus is represented in polynomial space ($\log(|p_i|)$ is polynomial).

At each step, the volume of the parallelepiped must at least halve, thus the volume at step t is $vol_t \leq |p_1| \times |p_2| \times \cdots \times |p_{d'}| \times \frac{1}{2^t}$. The procedure must saturate at or before the volume becomes 1, which occurs after at most

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

$\log(|p_1| \times |p_2| \times \cdots \times |p_{d'}|) = \sum_i \log(|p_i|)$ steps. At each step, we convert the $\mathbb{Z}$-linear set into Hermite normal form (a polynomial procedure), to retain exactly d' period vectors.

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

For all $x \in I$ and all alternate invariants I', $x \in I'$. By induction, let us prove that every invariant I must contain L_i. Clearly this is the case at L_0 because all points of $R_0 \subseteq \mathcal{O}$ must be in I and every period vectors in L_0 can be present, without loss of generality, due to Proposition 2. Assume $L_i \subseteq I$. Then it must be the case that I contains all points $M_j(L_i)$, as otherwise its not invariant. Since we now agree that I must contain all points in L_i and $M_j(L_i)$ we must agree that I contains L_{i+1}, as this is the minimal $\mathbb{Z}$-linear set covering L_i and $M_j(L_i)$ for all $j \leq k$. Since we take only the periods that this argument requires us to take, we have exactly the minimal invariant. □

Remark 1. Observe 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 smallest $\mathbb{Z}$-linear invariant.

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

In this section we show several generalisations of $\mathbb{Z}$-linear domain of invariants do not have strongest invariants.

$\mathbb{Z}$-semi-linear sets are the unions of $\mathbb{Z}$-linear sets, and therefore can include singleton sets. Consider the deterministic dynamical system starting from point 1 and doubling at each step $\mathcal{M} = (1, (x \mapsto 2x))$. This system has the 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 which is an improvement.

$\mathbb{N}$-linear sets generalise $\mathbb{Z}$-linear sets (observe $\mathbb{Z}$-linear sets are a proper subclass, since $\{x + p_i\mathbb{Z}\}$ can be expressed by $\{x + -p_i\mathbb{N} + p_i\mathbb{N}\}$, but $\mathbb{Z}$-linear cannot express $\{x + p_i\mathbb{N}\}$). Consider the LDS $((x_1, x_2), (\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}))$, with a reachability set consists 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

Up to now the invariants were for point targets. We now consider lattice-like targets, in particular targets specified as *full-dimensional* $\mathbb{Z}$-linear sets.

Theorem 3. *Given a dynamical system $(x^{(0)}, \{M_1, \dots, M_k\})$ it is decidable whether it 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, if it is unreachable, 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) [20]. However, even in the deterministic case if the linear-set is not required to be *full-dimensional* then the reachability problem is as hard as the Skolem problem (see, e.g. [22]): corresponding to the reachability of the target $\{(0, x_2, \dots, x_d) \mid x_2, \dots, x_d \in \mathbb{Z}\} = \{0 + e_2\mathbb{Z} + \cdots + e_d\mathbb{Z}\}$, 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 *full-dimension* linear sets can be expressed as 'square' hybrid-linear sets. Hybrid-linear sets are semi-linear sets where each of 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 all period vectors are the same multiple of a 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 $B \subseteq [0, m]^d$ such that $Y = \bigcup_{b \in B} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$.

Proof. Suppose $p_1, \dots, p_d$ form a d dimensional vector space. Let $P = \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix}$ be the matrix where the vectors p_i form the rows. Since P is full row rank then 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 integer for all j. Hence there is an integer combination of $p_1 \dots, p_d$ such that m_ie_i is an admissible direction in Y.

Let $m = \operatorname{lcm}\{m_1, \ldots, m_d\}$, and so 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 $x \in Y$ if and only $x' \in Y$ where $x'_i = (x_i \mod m)$.

And hence Y is equivalent to $\bigcup_{b \in B} \{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$ such that $B = \left\{x \in [0, m]^d \mid x \in Y\right\}$. □

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 $B' \subseteq [0, m]^d$.

We initialise the set $I_0 = \{x + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$ where $x \in [0, m]^d$ such that $x_j = (x_j^{(0)} \mod m)$. 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, $y \in [0, m]^d$ is formed by $y_j = ((M_i x)_j \mod m)$ for some $x \in I_0$. We iterate this constrution until it stabilizes in an inductive invariant I.

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, due to the modulo ste, result 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. It is clear, by construction that I is an inductive invariant avoiding the target. Observe there are at most $(m + 1)^d$ unique sets of the form $\{b + me_1\mathbb{Z} + \cdots + me_d\mathbb{Z}\}$, and so termination is also clear. □

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 invariant gain expressivity thanks to the 'direction' this is expressed by the period vector. 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, $\mathbb{N}$, can be expressed by $\mathbb{N}$-linear set invariants. We show that an $\mathbb{N}$-semi-linear invariant can *always* be found for a deterministic integer dynamical system, 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

In this subsection, we establish the following result

Theorem 4. *Given a deterministic LDS* $(x^{(0)}, M)$*, if the system does not reach a single vector target* y*, then an* $\mathbb{N}$*-semi-linear inductive invariant can be provided.*

To do so, we will use the results from [8] to compute an $\mathbb{R}_+$-semi-linear inductive invariant, then translate it into an $\mathbb{N}$-semi-linear inductive invariant. More precisely, the authors of [8] show how to build a polytope for some LDS (that potentially have non-integer points). Those polytopes are either bounded or are $\mathbb{R}_+$-semi-linear sets. In the first case, the set only contains finitely many integer points, each of which can be represented by an $\mathbb{N}$-linear set. In the second case, we build an $\mathbb{N}$-semi-linear containing exactly the set of integer points included in the $\mathbb{R}_+$-semi-linear set with 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, then one can build an* $\mathbb{N}$*-semi-linear set* N *representing every integer contained in* S*.*

Proof (Proof of Theorem 4). We remark that every invariant produced in [8] have rational period vectors as the vectors are given by the difference of successive point of the system and thus the result of Lemma 3 can always be applied. The authors of [8] build an inductive invariant in every case except when every eigenvalue of the matrix defining the evolution of the LDS is either 0 or of modulus 1 and at least one eigenvalue is not a root of unity. This case however cannot occur in our setting. Indeed, given a deterministic LDS $(x^{(0)}, M)$, M being an integer matrix, its roots are algebraic integers. For every root α of M an old result due to Kronecker [19] establishes that unless α is a root of unity, there exists another non-zero root of M (one of the Galois conjugate of α) which does not have modulus 1. This concludes the proof of Theorem 4. □

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

If the great expressivity of $\mathbb{N}$-semi-linear sets allows us to always find an invariant for deterministic LDS, it contributes in making the invariant synthesis problem undecidable when the LDS is not deterministic. We establish this undecidability result 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 $\{(u^1, v^1), \dots, (u^m, v^m)\}$ on alphabet $\{0, 2\}$, does there exist an infinite word $w = w_1 w_2 \dots$ on 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 equal to 8 [13,6].

Theorem 5. *The invariant synthesis problem for* $\mathbb{N}$*-semi-linear sets and LDS with at least 2 matrices of size* $7m + 35$ *is undecidable.*

Proof (Sketch). We establish the result for $m+5$ matrices of size 7. This can then be transformed in a usual way to two matrices of size $7m + 35$ (See Theorem 9 of [8] for instance).

Each pair of words of the infinite Post correspondence instance can be encoded as an integer: given a (finite or infinite) word w on the alphabet $\{0,2\}^*$, its quaternary encoding is $[w] = \sum_{i=1}^{|w|} w_i 4^{|w|-i}$. We select base 4 and digits $\{0,2\}$ instead of the canonical base 2 in order to have a 'sparse' encoding, which we need to define the invariants. One important property of this encoding is that we can add new letters to the encoded word simply by multiplying by a matrix.

If there is no positive answer to the ω-PCP instance, then the generated pair of words will differ at some point. This difference of letter create a gap in value that can be identified with an $\mathbb{N}$-semi-linear invariant. We therefore only have to register a finite number of points, the ones reached before the divergence of the words, plus the set of points where an important gap exists between the two encoded numbers.

If there is a positive answer, any $\mathbb{N}$-semi-linear invariant must contain the set of points obtained by encoding the solution of the instance. This set has no gap to register, and thus we establish that some points differing very slightly from the reachable points must be included in the invariant and using some additional matrices, we can reach the target from those points. Thus the invariant is not disjoint from the target. □

5.3 Nondeterministic one dimensional affine updates

The previous section shows that the nondeterministic case is undecidable once there are many dimensions, motivating analysis at lower dimensions. The MU Puzzle uses one dimension, with affine updates (equivalently two dimensions in Matrix representation, but one of the dimensions is always fixed to a constant 1). We consider this, one dimensional affine update case, and therefore, instead of the input consisting of matrices, we consider these replaced by affine functions of the form $f_i(x) = a_i x + b_i$.

Theorem 6. *Given $x^{(0)}, y \in \mathbb{Z}$, and a finite set of functions $\{f_1, \dots, f_k\}$ where $f_i(x) = a_i x + b_i$, $a_i, b_i \in \mathbb{Z}$ for $1 \le i \le k$, it is decidable whether y is reachable from $x^{(0)}$.*

If y is unreachable then an $\mathbb{N}$-semi-linear separating inductive invariant can be computed.

Decidability of the reachability problem is known [9,10], we extend 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 in the case of non-reachability will be an inductive invariant. In Section 6, we demonstrate the efficacy of our procedure with the implementation of a tool, which effectively tackles the MU Puzzle and its generalisation to arbitrary one-dimension affine functions.

Our case distinction will depend on the types 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 $f(x) = x + c$ 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.*

Proof. Firstly the identity function does nothing, so any trace with it is equivalent to the one without it. Suppose any other redundant function is used, then it is equivalent to doing so in the first step, or equivalently, using an alternative starting point. Hence the problem can be reduced to finitely many instances of the problem with different starting points. Take the union of the resulting invariants. □

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 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 two unique pure inverting transitions. 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 counter, we essentially move in either direction by some fixed value. This means we can be sure that we can use only $\mathbb{Z}$-linear invariants, rather than requiring $\mathbb{N}$-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 reachable sets using the following lemma:

Lemma 8. *Let $h(x) = x + d$ be chosen 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 $x \in I$, then $f(x) = ax + b \in I$. Indeed, consider $x + dk \in \{x + d\mathbb{Z}\}$ then $f(x + dk) = ax + adk + b = f(x) + adk$ must be in the invariant.

However if some counter $h(x) = x + d$ exists then by choosing the initial $k \in \mathbb{Z}$ appropriately and applying $h(x)$ $m \in \mathbb{N}$ times one can reach $f(x) + adk + dm = f(x) + dn$ for $n \in \mathbb{Z}$. □

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

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

Only pure inverters If there is exactly one pure inverter $f(x) = -x + b$, then $f(x^{(0)}) = -x^{(0)} + b$ and $f(-x^{(0)} + b) = x^{(0)} - b + b = x^{(0)}$, then the set of reachable points is finite (invariant $\left\{x^{(0)}, -x^{(0)} + b\right\}$).

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

Lemma 9. *For every $M \ge 0$ and every growing function $f(x) = ax + b$, $|a| \ge 2$, there exists $C_f^M \ge 0$ s.t. if $|x| \ge C_f^M$ then $|f(x)| \ge |x| + M$.*

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

This is the only case where the invariant is not exactly the reachability set, however, we find a window where we compute everything, and then from some point on, we take everything.

Let $C = \max\left\{C_{f_1}^0, \ldots, C_{f_k}^0 y\right\}$, 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 use the inductive invariant $\{-C - \mathbb{N}\} \cup \{C + \mathbb{N}\} \cup (\mathcal{O} \cap [-C, C])$. The set $\mathcal{O} \cap [-C, C]$ (a finite set of singleton points) can be computed by total exploration of the points (e.g. by breadth first search) until points are larger than C (since then they are forever outside of 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, and (b) an inverting growing function only ensures that $-C$ is mapped to $C + 1$, rather than $C + d$. Hence, we choose C' to ensure $C \pm d$ is still growing by at least d. Let $C' = \max\left\{C_{f_1}^{|d|}, \ldots, C_{f_k}^{|d|}, y\right\} + |d|$. 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 case (if there are not two opposing counters, and not all functions are growing or pure inverters), then there are counter-like functions, but they are all counting in the same direction. There may also be a single inverter, and there may be growers too.

Choose a single counter $h(x) = x + d$ to be the reference counter, the choice is arbitrary, so one may as well choose $h(x)$ minimising $|d|$. As a starting point, we have $\left\{x^{(0)} + d\mathbb{N}\right\} \subseteq I$.

Lemma 10. *If there is a pure 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 between Lemma 8 and here is the observation that now a $\mathbb{N}$-linear set has induced a $\mathbb{Z}$-linear set.

Proof. Let $r = g(x) + dm$ for $m \in \mathbb{Z}$, we show $r \subseteq I$. 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 $h(x)$ k-times. Hence for any $m \in \mathbb{Z}$ there exists $k, n \in \mathbb{N}$ s.t. $k + -na = m$, thus r is reachable. □

Similarly to the two opposing counter case, whenever there is a $\mathbb{Z}$-linear set in the invariant, Lemma 8 allows us to saturate the amongst the reachable modulo classes. There are finitely many such modulo classes and this will terminate.

However, not every set will be an $\mathbb{Z}$-linear set. For example there can be no inverter, or some modulo classes are unreachable after applying an inverter.

Consider $\{x + d\mathbb{N}\} \in I$, which is not yet invariant. we repeatedly apply non-inverting functions to $\{x + d\mathbb{N}\}$ we 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 opposing directions, we may generate infinitely many such classes.

Example 3. Consider the reference counter $h(x) = x + 4$, with the initial point 5. Together these imply 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.

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 thus 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 initially moving with the direction 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 -26 is as well). This means in the negative direction we can obtain an arbitrarily large negative value 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 observe all reachable modulo classes, and any modulo class reachable after an inverting function induces an$\mathbb{Z}$-linear sets. So it remains to consider consider which $\mathbb{N}$-linear sets are reachable without inverting functions. Intuitively the remaining case occurs when we saturate $\mathbb{N}$-linear sets until they repeat modulo class modulo d in the direction opposing 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 we a sets $\{x + d\mathbb{N}\}$ leads to $\{y + d\mathbb{N}\}$ by a sequence of non-inverting functions with $x \cong y \mod d$. Then by the 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 formulate the formal proof in Appendix E.2.

Reachability The above procedure is sufficient to decide reachability. In all cases except the 'no counter case' the invariants described exactly define all reachable points and reachability can be detected by asking if the target is in the invariant.

In the remaining case, the invariant 'depends' on the target, which induces the bound C'. The target is within the region $[-C', C' + d]$, in which every reachable point was computed anyway, and so the target is reachable only if its in the invariant returned by the procedure. However, for a target of larger modulus a new invariant would need to be built.

Complexity

Lemma 11. *Assume 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 may be of exponential size, and thus it would take at least exponential time to compute. That is because the invariant we construct may includes every value in an interval, for example, $[-C, C]$, where C is of polynomial size in the largest value.

As shown by [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 packed any number of times rather than 0 or 1). Moreover the Knapsack problem is efficiently solvable in pseudo-polynomial time by dynamic programming; that is, polynomial time assuming the input is in unary, which we also have for our procedure.

In practice this can mean the algorithm is efficient on a large and practical class of inputs. For example the input size may be very small; expressing $f(x) = 2x + 200$ needs approximately only $\log(2) + \log(200) \approx 10$ bits, but one realistically must do work proportional to 200, which can still be reasonable.

6 Tool

Our invariant synthesis tool computes $\mathbb{N}$-semi-linear invariants for point and $\mathbb{Z}$-linear targets on systems defined by one dimensional affine functions. That is the procedures, of Theorem 3 (restricted to one dimensional affine) and Theorem 6. The tool is built in python and can be used by command line file input, a web interface or by directly invoking the python packages. The output on the MU Puzzle can be seen in Appendix F.

The tool takes as input an instance (a start point, a target and a series of functions) and returns the generated invariant. Additionally it proves this invariant is indeed invariant: the invariant is a union of linear sets, so for each

Size	Invariant Build Time		Unreachable Instances	Invariant Proof Time		Reachable Instances	Reachable with proofs within ≈30s	Reachability Proof time
	avg	max		avg	max			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

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 30s due to instances that terminate just before the timeout).

linear set and each function, the tool demonstrates the application of that function to the linear set and shows for which linear set in the invariant this is a subset. Using this invariant it can decide reachability; if the system is reachable the invariant is not in itself a proof of reachability (the set $\{0 + \mathbb{Z}\}$ is invariant for any instance, but does not prove reachability and is clearly often wrong). Instead, equipped with the assurance of reachability the system searches for a direct proof of reachability: a sequence of function from start to target (a process which would not otherwise be guaranteed to terminate).

Experimentation The tool 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 1 of each function type).

Our analysis, depicted in Table 2 demonstrates the effects 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 the system successfully built the invariant, and hence decided reachability, very quickly (on average well below 1 second) and also produced the proof of invariance in around half a second on average. To demonstrate correctness in instances where the target is reachable the system also attempts to produce a proof of reachability (a sequence of functions from start to target). Since our paper is focused on invariants to demonstrate non-reachability, this proof of reachability procedure was implemented only via a simple breadth-first search without any heuristics and so a timeout of 30 seconds was used for this part of the experiment only.

Our experimental methodology is partially limited due to the high likelihood of a reachable instance. A random instance is likely to lead to the either the whole space or a large portion of the space being reachable, so in particular, a randomly chosen target is highly likely to be reachable. If two random counters are included, the chance that $\gcd(b_1, b_2) = 1$ (hence the whole space covered) is around 60.8% and higher if more counters are chosen.

Overall around 86% of instances were reachable instances (of which 84% produced a proof within 30 seconds). Of the 14% of unreachable instances, all produced a proof, with the invariant taking around 0.2 seconds to build and 0.6 seconds to produce the proof. The 30 seconds timeout when demonstrating reachability directly is several orders of magnitudes longer than answering the reachability question 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). However, given the absence of comparable benchmarks, we do not claim to be providing accurate analysis of the specific running time, but instead believe our analysis depicts behaviour which is more than reasonable in practice on reasonably sized instances.

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 them 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.

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.

References

1. Almagor, S., Chistikov, D., Ouaknine, J., Worrell, J.: O-minimal invariants for discrete-time dynamical systems. preprint, submitted (2019)
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, July 15-19, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6174, pp. 227–242. Springer (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. Journal of Soviet Mathematics **34**(4), 1838–1882 (1986)
4. Clarke, E.M., Fehnker, A., Han, Z., Krogh, B.H., Ouaknine, J., Stursberg, O., Theobald, M.: 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 (1979)
8. Fijalkow, N., Lefaucheux, E., Ohlmann, P., Ouaknine, J., Pouly, A., Worrell, J.: On the monniaux problem in abstract interpretation. In: Chang, B.E. (ed.) Static Analysis - 26th International Symposium, SAS 2019, Porto, Portugal, October 8-11, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11822, pp. 162–180. Springer (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.) 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, pp. 409–420. Springer (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., 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.: Analyzing program termination and complexity automatically with aprove. J. Autom. Reason. **58**(1), 3–31 (2017). https://doi.org/10.1007/s10817-016-9388-y
12. Ginsburg, S., Spanier, E.H.: Bounded algol-like languages. Transactions of the American Mathematical Society **113**(2), 333–368 (1964), http://www.jstor.org/stable/1994067
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.) 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, pp. 797–813. Springer (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, July 09-12, 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. Markov, A.: On certain insoluble problems concerning matrices. In: Doklady Akad. Nauk SSSR. vol. 57, pp. 539–542 (1947)
21. Monniaux, D.: On the decidability of the existence of polyhedral invariants in transition systems. Acta Informatica **56**(4), 385–389 (2019). https://doi.org/10.1007/s00236-018-0324-y
22. Ouaknine, J., Worrell, J.: 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, pp. 21–28. Springer (2012). https://doi.org/10.1007/978-3-642-33512-9_3
23. 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)
24. 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

A Proof of Lemma 1

Lemma 1 (proof in Appendix A). *Given an LDS $(x^{(0)}, \{M_1, \dots, M_k\})$ of dimension d, we can compute in time polynomial in d, k and $\log \mu$ (where $\mu > 0$ is an upper bound of the absolute values of the integers appearing in $x^{(0)}$ and $M_1, \dots, M_k$), a $\mathbb{Q}$-affinely independent set of 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 $\mu_0 := (d\mu)^d$.*

Proof. The result of [24, 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 of the current version this set. Whilst the result of [24] 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 its children are linearly independent of 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 vector is reached with at most d applications of matrices to $x^{(0)}$. Hence, if the largest absolute value of a point is μ, after d iterations, the largest absolute value is $(d\mu)^d$. This is by induction on the largest possible value μ for every entry:

$$\text{Base case:} \begin{pmatrix} \mu \dots \mu \\ \ddots \\ \mu \dots \mu \end{pmatrix} \begin{pmatrix} \mu \\ \vdots \\ \mu \end{pmatrix} = \begin{pmatrix} d\mu \\ \vdots \\ d\mu \end{pmatrix}$$

$$\text{Inductive case:} \begin{pmatrix} \mu \dots \mu \\ \ddots \\ \mu \dots \mu \end{pmatrix} \begin{pmatrix} (d\mu)^k \\ \vdots \\ (d\mu)^k \end{pmatrix} = \begin{pmatrix} d(\mu(d\mu)^k) \\ \vdots \\ d(\mu(d\mu)^k) \end{pmatrix} = \begin{pmatrix} (d\mu)^{(k+1)} \\ \vdots \\ (d\mu)^{(k+1)} \end{pmatrix}$$

The result of [24] is in polynomial time in the number of arithmetic operations, we observe that this is also polynomial time in the bit-size. The independence checking in the algorithm involves checking linear independence of at most d vectors all having bit size at most $log((d\mu)^d) = d\log(d) + d\log(\mu)$, which can be done in polynomial time in the bit-size (for example by Bareiss algorithm for calculating the determinant). □

B Proof of Theorem 1

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

Proof (Proof of Theorem 1). Given $A_1, \dots, A_k$ such that $\overline{\mathcal{O}}^z = A_1 \cup \dots \cup A_k$, Proposition 1 tells us we need to compute $\overline{A_i}^a$'s. This can be done using Karr's [17] algorithm.

It remains to confirm that Karr's algorithm can be applied to an algebraic set A. Start with $\overline{A}^a \leftarrow \{x\}$ for some point $x \in A$. Repeatedly let $\overline{A}^a \leftarrow \overline{\overline{A}^a \cup \{y\}}^a$ where $y \in A \setminus \overline{A}^a$. Such a point can be found using quantifier elimination in the theory of the reals. Each step necessarily increases the dimension, which can occur at most d times, entailing termination. □

C Proof of Lemma 3

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, then one can build an $\mathbb{N}$-semi-linear set N representing every integer contained in S.*

Proof. Let $S = \{x + \sum_i p_i \mathbb{R}_+\}$ be a $\mathbb{R}_+$-linear set where the vectors p_i have rational coefficients and x is an integer vector. Let $k \in \mathbb{N}$ so that the vectors kp_i have integer coefficients. We denote by v_i the integer vectors obtained as a convex combination of the vectors kp_i. Then the set $T = \{x + \sum_i v_i\mathbb{N}\}$ contains exactly the integer vectors contained in S.

Indeed, first T only contains integer points as both x and the vectors v_i are integer vectors. Secondly, all the vectors in T are included in S as the period vectors of T lie in the convex hull of the vectors of S. Finally, given an integer vector y in S, y can be rewritten as $y = x + v + \sum_i m_i kp_i$ where for all $i, m_i \in \mathbb{N}$ and v is an integer vector lying in the convex hull of the vectors kp_i. Therefore there exists j such that $v_j = v$ and as for all i, kp_i is a period vector of T, $y \in T$. □

D Proof of Theorem 5

Theorem 5. *The invariant synthesis problem for* $\mathbb{N}$*-semi-linear sets and LDS with at least 2 matrices of size* $7m + 35$ *is undecidable.*

Proof. We will prove the result for $m + 5$ matrices of size 7. This can then be transformed in a usual way to two matrices of size $7m+35$ (See Theorem 9 of [8] for instance).

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 p this dimension, it is initially equal to 1 and its target value is 0. Consider the three following affine transformation: $f_1(p) = 2p - 1$, $f_2(p) = 2p - 2$ and $f_3(p) = 2p$, then the only sequences of transformation 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 $\mathcal{I}$ ensure that any sequence of transformation 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 implictly contain the function f_1, f_2 or f_3.

We reduce an instance $\{(u^1, v^1), \ldots, (u^m, v^m)\}$ of the ω-PCP problem to the invariant synthesis problem. In order to simplify future notations, 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 $\{1, \ldots, m\}$ we denote by u^w and v^w the words on the alphabet $\{0, 2\}$ such that $u^w = u^{w_1} u^{w_2} \ldots$ and $v^w = v^{w_1} v^{w_2} \ldots$. Given a (finite or infinite) word w on the alphabet $\{0, 2\}^*$, 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 $\max_i = \max(n_i, m_i)$.

We work with 5 dimension, (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) encode the action of reading the pair (u^i, v^i) and increases the counters n and k: it simultaneously applies $s \leftarrow \max_i s + [u^i]\frac{\max_i}{n_i} - [v^i]\frac{\max_i}{n_i}$, $c \leftarrow \frac{\max_i}{n_i}c$, $d \leftarrow \frac{\max_i}{n_i}d$, $n \leftarrow n + k$ $k \leftarrow k + 1$.

- The type 2 transformation $\mathsf{Transfer}$ on (s, c, d, n, k) gather some of the values in order to compare them: $s \leftarrow s - c - d$, $c \leftarrow -s - c - d$.
- The type 3 transformation $\mathsf{Inc_s}$ increments s: $s \leftarrow s + 1$.
- The type 3 transformation $\mathsf{Inc_c}$ increments u: $c \leftarrow c + 1$.
- 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 need 7 dimensions in total: the five above, (s, c, d, n, k), the one used for ordering the transformations,p, and one dimension constantly equal to 1, required to use affine transformations.

We now show that there is a solution to the given instance of the ω-PCP problem iff there does not exist a $\mathbb{N}$-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.

Assume first 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, $x_j = (s_j, c_j, 0, n_j, k_j, 0, 1) = \mathsf{Transfer}\ \mathsf{Simulate}_{\mathsf{w}_{\leq \mathsf{j}}} x$ where $\mathsf{Simulate}_{\mathsf{w}_{\leq \mathsf{j}}}$ represents the transformation $\mathsf{Simulate}_{\mathsf{w_j}} \ldots \mathsf{Simulate}_{\mathsf{w_2}} \mathsf{Simulate}_{\mathsf{w_1}}$. We have that s_j and c_j are negative. Indeed, let (s, c, d) be the three first components of $\mathsf{Simulate}_{\mathsf{w}_{\leq \mathsf{j}}} x$, we have that $s = c[u^{wi}] - d[v^{wi}]$. As $w_{\leq j}$ is a prefix of a solution to the ω-PCP instance, assuming $|u^w i| \leq |v^w i|$ this implies that

$$\begin{aligned} |s| &= |c[u^{wi}] - d[v^{wi}]| \\ &= \sum_{j=1}^{|v^w i|} |u_j^{wi} - v_j^{wi}| c4^{|u^w i| - j} \\ &= \sum_{j=|u^w i|+1}^{|v^w i|} v_j^{wi} c4^{|u^w i| - j} \\ &\leqslant \frac{2c}{3} \end{aligned}$$

Thus $|s| - c - d$ is negative, thus $s_j = s - c - d$ and $c_j = -s - c - d$ are negative.

Due to the above, by applying to the points x_j a number of time the transformations $\mathsf{Inc_s}$ and $\mathsf{Inc_c}$, we obtain the sequence of points (y_j) 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 shape $(0, 0, 0, 0, n_j + d, k_j, 0, 1)$ 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 point (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). 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) 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 $\mathbb{N}$-semi-linear set over-approximating the system intersects with the target.

Conversely, 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 $\{0, \ldots, m\}$ there exists $n \leq n_0$ such that $u_n^w \neq v_n^w$. Indeed, consider the tree which 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 $i = 1 \ldots m$. 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

$$\mathcal{I}_1 = \big\{\mathsf{Simulate_w}(x) \mid w \in \{1, \ldots, m\}^* \wedge |w| \leq n_0 + 1\big\},$$

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

and

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

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.

• Let $z = \mathsf{Simulate_w}(x) \in \mathcal{I}_1$, for some $w \in \{1, \ldots, m\}^*$ 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 $i \leq m$, if $|w| \leq n_0$, then $\mathsf{Simulate_i} z \in \mathcal{I}_1$. Else, $\mathsf{Simulate_{wi}} x = (s, c, d, n, k, p, 1)$ 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

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

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

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

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

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

• 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 is 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 $d' \geqslant c'$ (this is completely symmetric in c' and d'). We have that

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

since $\max_i c \geq c'$, $\max_i d/3 \geq d'$ (as $m_i \geq 4$) and $\max_i \geqslant 4$. 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. □

E Additional proofs for Theorem 6

E.1 Proof of Lemma 7

Lemma 12. *For ℓ, k coprime, the sequence $a_n = (n\ell \mod k)$ for $n \in \mathbb{N}$ cycles through every modulo 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 $\gcd(k, \ell) = 1$. □

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

Proof. 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 12 there exists n such that $n\ell$ is in any modulo class modulo k, and thus too for $1 + n\ell$ and so k divides $1 + n\ell$ for some n.

Hence the set $\{x + d\mathbb{N}\}$ is included in the reachability set: we obtain $x + jd$ by applying function f mj times and applying function g nj times. Similarly, we can find $m', n' \geq 0$ such that $m'b - cn' = -d$ and thus $\{x + d\mathbb{Z}\}$ is within the reachability set. □

E.2 Extended argument for non opposing counters

The following shows that if $\{x + d\mathbb{N}\}$ does lead to $\{y + d\mathbb{N}\}$, with $y < x$ and $y \cong x \mod d$, then indeed we can reach $\{z + d\mathbb{N}\}$ for any $z \cong x \mod d$ by reapplying the same set of functions which lead from x to y.

Lemma 13. *Assume the reference counter $h(x) = x + d$ with $d \geq 0$. Suppose all growing functions are growing outside of $[-B, C]$. Consider $x^{(0)} \in I$ and a path $x^{(0)}, f_{i_1}, x^{(1)}, f_{i_2}, \ldots, f_{i_m}, x^{(m)}$ such that $x^{(j)} = f_{i_j}(x^{(j-1)})$, $x^{(j)} \leq -B$, $x^{(m)} < x^{(0)}$ and $x^{(0)} \cong x^{(m)} \mod d$.*

Then $\left\{x^{(0)} + d\mathbb{Z}\right\} \subseteq I$.

Proof. The re-application of $f_{i_m} \circ \cdots \circ f_{i_1}$ results in the same modulo class by modulo arithmetic. Further since $x^{(j)} \leq -B$ then any growing f_{i_j}, is growing by at least as much as in the first application. Thus $f_{i_m} \circ \cdots \circ f_{i_1}(x^{(m)}) < x^{(m)}$.

Hence for any $M < -B$, there exists $c < M, n \in \mathbb{N}$, such that $(f_{i_m} \circ \cdots \circ f_{i_1})^n(x^{(0)}) = c \cong x^{(0)} \mod d$. Hence for any $x^{(0)} + kd \in \left\{x^{(0)} + d\mathbb{Z}\right\}$ with $k \in \mathbb{Z}$ then there exists n, ℓ such that $(f_{i_m} \circ \cdots \circ f_{i_1})^n(x^{(0)}) \leq x^{(0)} + kd$ and $h^\ell \circ (f_{i_m} \circ \cdots \circ f_{i_1})^n(x^{(0)}) = x^{(0)} + kd$. □

Remark 3. By symmetry, Lemma 13 also holds for the opposite direction. That is when $h(x) = x - d$, $d > 0$, inequalities are inverted and C is used in place of $-B$.

We now consider inductively applying non-inverting functions to sets $\{x + d\mathbb{N}\} \in I$. Then add $\{f_i(x) + d\mathbb{N}\}$ provided it is not already a subset of some set already in I. If $\{f_i(x) + d\mathbb{N}\}$ is new and a new modulo class we can again apply Lemma 10, from whence we may also need to apply Lemma 8.

However, when this procedure does not saturate there eventually exists be a sequence of actions in which $\{x + d\mathbb{N}\}$ leads to $\{y + d\mathbb{N}\}$ with $x \cong y \mod d$ according to a path in Lemma 13. In particular $y < x < -B$ since if $x < y$ then $\{y + d\mathbb{N}\} \subseteq \{x + d\mathbb{N}\}$, some modulo class must repeat after at most d steps, and eventually the procedure must stay $< -B$ for at least d steps. Then, according to Lemma 13, a new $\mathbb{Z}$-linear set can be added ($\{x + d\mathbb{Z}\}$) (which again can be saturated using Lemma 8). We repeat this process until all $\mathbb{N}$-linear sets are invariant. This process terminates, as each application of Lemma 13 adds a new $\mathbb{Z}$-linear set with period d, of which there are at most d.

E.3 Proof of Lemma 11

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

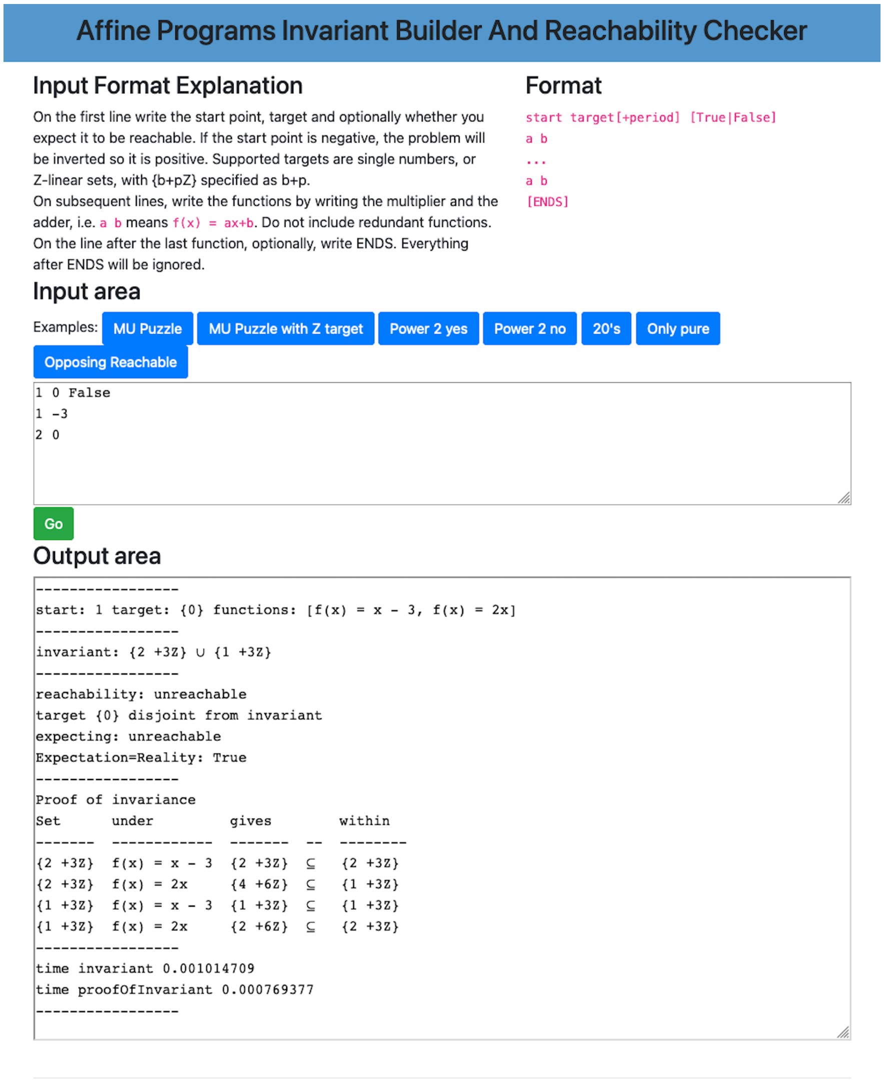
Proof. In the no-counter case, by Lemma 9, there is an interval $[-C, C]$ of size approximately $\frac{|b|+|M|}{|a|-1}$, where $|b|, |M|, |a|$ are all numbers represented in the input, and thus is of polynomial of size. This means the gap is of polynomial size, and thus the saturation algorithm, which must in each step add a point or terminate, is of polynomial time.

In each counter-case there is a reference counter period d arising directly from the input as the counter part of some function, or in the case of two opposing counters, possibly the sum of two counter parts. For this period d there are at most $3d$ possible types of non-singleton invariant ($\{x + d\mathbb{N}\}$ or $\{x - d\mathbb{N}\}$ for some x and $x + d\mathbb{Z}$ for $x \in \{0, \ldots, d\}$). Singletons only arise in the interval $[-C, C]$ if they exist. Hence, there are at most $O(2C + 3d)$ steps which change the invariant. In the case of invariants induced by Lemma 13, there can be a path of length at most d steps outside of $[-C, C]$ before a cycle is found. □

F Tool Web Interface

The tool's web interface, applied to the MU Puzzle can be seen in the following image, where the system has induced the invariant $\{2+3\mathbb{Z}\} \cup \{1+3\mathbb{Z}\}$.

The web-interface can be tested at http://invariants.1251.uk[9].



Affine Programs Invariant Builder And Reachability Checker

Input Format Explanation

On the first line write the start point, target and optionally whether you expect it to be reachable. If the start point is negative, the problem will be inverted so it is positive. Supported targets are single numbers, or Z-linear sets, with {b+pZ} specified as b+p.
On subsequent lines, write the functions by writing the multiplier and the adder, i.e. `a b` means `f(x) = ax+b`. Do not include redundant functions.
On the line after the last function, optionally, write ENDS. Everything after ENDS will be ignored.

Format

```
start target[+period] [True|False]
a b
...
a b
[ENDS]
```

Input area

```
1 0 False
1 -3
2 0
```

Output area

```
-----------------
start: 1 target: {0} functions: [f(x) = x - 3, f(x) = 2x]
-----------------
invariant: {2 +3ℤ} ∪ {1 +3ℤ}
-----------------
reachability: unreachable
target {0} disjoint from invariant
expecting: unreachable
Expectation=Reality: True
-----------------
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}
-----------------
time invariant 0.001014709
time proofOfInvariant 0.000769377
-----------------
```

[9] Anonymisation provided on a best effort basis.