Closed Forms for Numerical Loops[*]

ZACHARY KINCAID, Princeton University, USA
JASON BRECK, University of Wisconsin, USA
JOHN CYPHERT, University of Wisconsin, USA
THOMAS REPS, University of Wisconsin, USA and GrammaTech, Inc., USA

This paper investigates the problem of reasoning about non-linear behavior of simple numerical loops. Our approach builds on classical techniques for analyzing the behavior of linear dynamical systems. It is well-known that a closed-form representation of the behavior of a linear dynamical system can always be expressed using algebraic numbers, but this approach can create formulas that present an obstacle for automated-reasoning tools. This paper characterizes when linear loops have closed forms in simpler theories that are more amenable to automated reasoning. The algorithms for computing closed forms described in the paper avoid the use of algebraic numbers, and produce closed forms expressed using polynomials and exponentials over rational numbers. We show that the logic for expressing closed forms is decidable, yielding decision procedures for verifying safety and termination of a class of numerical loops over rational numbers. We also show that the procedure for computing closed forms for this class of numerical loops can be used to over-approximate the behavior of arbitrary numerical programs (with unrestricted control flow, non-deterministic assignments, and recursive procedures).

CCS Concepts: • **Theory of computation** → **Program analysis**; *Logic and verification*; • **Computing methodologies** → *Symbolic and algebraic algorithms*;

Additional Key Words and Phrases: Invariant generation, loop summarization, decision procedures

ACM Reference Format:
Zachary Kincaid, Jason Breck, John Cyphert, and Thomas Reps. 2019. Closed Forms for Numerical Loops. *Proc. ACM Program. Lang.* 3, POPL, Article 55 (January 2019), 29 pages. https://doi.org/10.1145/3290368

1 INTRODUCTION

Many programs exhibit non-linear behavior, whether explicitly—e.g., scientific or cyber-physical applications—or implicitly—e.g., time or space usage of nested loops or recursive procedures. This paper addresses a problem in the basic science of program analysis: *how can we systematically (i.e., rather than heuristically) reason about non-linear behavior?* We consider a simplified model of numerical loops with linear and polynomial assignments. We identify conditions under which it is possible to exactly characterize the behavior of such a loop with a logical formula involving

[*] This work was supported in part by a gift from Rajiv and Ritu Batra; by AFRL under DARPA MUSE award FA8750-14-2-0270 and DARPA STAC award FA8750-15-C-0082; by ONR under grant N00014-17-1-2889; and by the UW-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from WARF. Opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

Authors' addresses: Zachary Kincaid, zkincaid@cs.princeton.edu, Princeton University, Princeton, NJ, USA; Jason Breck, jbreck@wisc.edu, University of Wisconsin, Madison, WI, USA; John Cyphert, jcyphert@wisc.edu, University of Wisconsin, Madison, WI, USA; Thomas Reps, reps@cs.wisc.edu, University of Wisconsin, Madison, WI, USA, GrammaTech, Inc. Ithaca, NY, USA.

2475-1421/2019/1-ART55
https://doi.org/10.1145/3290368

```
while (*) do
    int $x_{\text{old}} = x$;
    int $y_{\text{old}} = y$;
    $x = 2x_{\text{old}} + y_{\text{old}}$;
    $y = x_{\text{old}} + 2y_{\text{old}}$;
```
(a)

```
while (*) do
    $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
```
(a')

```
while (*) do
    int $x_{\text{old}} = x$;
    $x = y$;
    $y = -x_{\text{old}}$;
```
(b)

```
while (*) do
    $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
```
(b')

Fig. 1. Examples used to illustrate the challenges of finding summaries of linear loops.

exponenentials and polynomials over the rationals, and show that this logical fragment is decidable. As a consequence, we obtain decidability results for safety and termination problems for a simple program model that can exhibit non-linear behavior.

Example 1.1. The loops shown in Fig. 1 (a) and (b) are *linear loops*: non-deterministic loops that consist of a sequence of affine assignments—or, equivalently loops that can be written in the form **while** $(*)$ **do** $\{ \mathbf{x} = A\mathbf{x} \}$ (Fig. 1 (a') and (b')), where $(*)$ denotes a non-deterministic exit condition. In loop (a), the values of x and y produce the following sequence, as a function of their initial values x_0 and y_0:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} 2x_0 + y_0 \\ x_0 + 2y_0 \end{pmatrix}, \begin{pmatrix} 5x_0 + 4y_0 \\ 4x_0 + 5y_0 \end{pmatrix}, \begin{pmatrix} 14x_0 + 13y_0 \\ 13x_0 + 14y_0 \end{pmatrix}, \dots$$

The behavior of linear loops is well-studied in the field of dynamical systems (and in program analysis—see, e.g., analysis of termination of linear loops in [Braverman 2006; Tiwari 2004] and acceleration of linear loops in [Boigelot 2003; Jeannet et al. 2014]). The classical method for obtaining a closed-form representation of the behavior of a linear loop of the form **while** $(*)$ **do** $\{ \mathbf{x} = A\mathbf{x} \}$ is by symbolically exponentiating the matrix A (see §3 for more information). Using symbolic matrix exponentiation, we can characterize the values of x and y that arise at the head of the loop—and thus also the values on exit from the loop—as a function of the number of iterations k via the following formula:

$$\left(x' = (3^k + 1)x_0/2 + (3^k - 1)y_0/2\right) \land \left(y' = (3^k - 1)x_0/2 + (3^k + 1)y_0/2\right)$$

Now consider loop (b). In (b), x and y produce the following sequence:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} y_0 \\ -x_0 \end{pmatrix}, \begin{pmatrix} -x_0 \\ -y_0 \end{pmatrix}, \begin{pmatrix} -y_0 \\ x_0 \end{pmatrix}, \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \dots$$

Symbolic matrix exponentiation yields the following formula that captures the behavior of this loop:

$$\begin{aligned} & \left(x' = x_0(i^k/2 + (-i)^k/2) + y_0((-i)i^k/2 + i(-i)^k/2)\right) \\ \land \quad & \left(y' = x_0(i^k/2 - i(-i)^k/2) + y_0(i^k/2 + (-i)^k/2)\right). \end{aligned} \tag{1}$$

Notice that this formula makes use of the imaginary unit i: powers of i and $-i$ are used as a kind of switching network to include/exclude x_0 and y_0 for selected powers of k. □

Classical symbolic matrix exponentiation produces a closed-form formula that involves polynomials and exponentials over the eigenvalues of the matrix for the loop. In general, these eigenvalues are algebraic numbers. For instance, the eigenvalues of the matrix for loop (b) are i and $-i$ (see §3), and the closed-form representation is Eqn. (1). Unfortunately, exponential-polynomial expressions over algebraic numbers are difficult to reason about. For instance, the problem of determining whether such an expression has a root in the natural numbers is equivalent to Skolem's problem for linear recurrence sequences. The question of whether that problem is decidable has been open since the 1930s [Ouaknine and Worrell 2015].

An alternative to symbolic matrix exponentiation is given in [Kincaid et al. 2018]. Kincaid et al. [2018] express a closed-form representation of linear loops using additional function symbols in

```
for (i = 0; i < N; i++)
   for (j = 0; j < N; j++)
      C[i][j] = A[i][j] + B[i][j];
```

(*a*)

while $(*)$ **do**

$$\begin{bmatrix} i \\ j \\ a \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ a \\ n \\ 1 \end{bmatrix}$$

(*b*)

while $(*)$ **do**

$$\begin{bmatrix} i \\ j \\ a \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ a \\ n \\ 1 \end{bmatrix}$$

(*c*)

Fig. 2. A nested loop that exhibits non-linear behavior

place of exponentials of algebraic numbers. This approach is advantageous because it enables *heuristic* reasoning about non-linear behavior using SMT solvers (treating the additional function symbols as uninterpreted function symbols), but does not allow *systematic* reasoning: if the function symbols are interpreted, then the logic is just as expressive as exponential-polynomial arithmetic over algebraic numbers, and suffers from the same intractability.

This paper gives conditions under which a closed-form representation of a loop can be expressed in weaker logics that are more amenable to automated reasoning. In particular, we seek closed forms in decidable logics that avoid the use of algebraic numbers. For instance, our method produces an alternative closed-form representation for loop (b) by making a case distinction on whether the loop iteration is even or odd:

$$\begin{array}{rl} & \left(k \equiv 0 \bmod 2 \land x' = (-1)^{\lfloor k/2 \rfloor} x_0 \land y' = (-1)^{\lfloor k/2 \rfloor} y_0\right) \\ \lor & \left(k \equiv 1 \bmod 2 \land x' = (-1)^{\lfloor k/2 \rfloor} y_0 \land y' = -(-1)^{\lfloor k/2 \rfloor} x_0\right). \end{array} \tag{2}$$

Although the logical fragment we use to express closed forms of loops is weaker than exponential-polynomial arithmetic over algebraic numbers, it is still very expressive, allowing us to capture polynomial and exponential behavior. We show that, despite the high degree of expressivity, the satisfiability problem for this logic is decidable. As a consequence, we obtain decision procedures for problems related to safety and termination of linear loops that meet certain efficiently checkable technical conditions (to be described in §5). For instance, we can automatically prove the validity of the Hoare triple "$\{x = 1 \land y = 1\}$ Fig. 1(b) $\{x \leq 1\}$" by proving that the formula "$x = 1 \land y = 1 \land$ Eqn. (2) $\land\, x' > 1$" is unsatisfiable.

Although our concern in this paper is with a simplified program model, using the abstraction techniques of Kincaid et al. [2018, §5] our results have immediate applications, as illustrated in the following example.

Example 1.2. Consider the matrix addition routine in Fig. 2(a). Suppose that we wish to count the number of memory accesses made by this routine. By introducing (in the innermost loop) a synthetic variable a that is incremented by 3 (the number of memory accesses in one iteration in the innermost loop), we can extract (automatically, using [Kincaid et al. 2018, §5]) the linear model of the inner loop shown in Fig. 2(b). The closed form we compute of the inner loop is

$$\exists k \in \mathbb{N}. i' = i \land j' = j + k \land a' = a + 3k \land n' = n\,,$$

which combined with the precondition and post-condition of the innermost loop (see §6.3) yields the following representation of the action of the innermost loop:

$$\exists k \in \mathbb{N}. i' = i \land j' = n \land a' = a + 3n \land n' = n\,.$$

Again employing the abstraction technique of [Kincaid et al. 2018, §5] (and using the above formula to summarize the inner loop), we extract the linear model of the outer loop shown in Fig. 2(c), and compute the closed form

$$\exists k \in \mathbb{N}. i' = i + k \land j' = n \land a' = a + 3n^2 \land n' = n\,, \tag{3}$$

from which we see that the number of memory accesses in the addition routine is exactly $3n^2$. □

Contributions. Our work makes contributions in three main areas:

(1) We present algorithms that solve the problem of obtaining—in a decidable logic—closed-form formulas of the kind given in Eqns. (2) and (3), namely, loop summaries that capture the iterated behavior of a linear loop (or an over-approximation thereof).
   - We observe that if a matrix has *periodic rational* eigenvalues (i.e., each eigenvalue is an n^{th} root of a rational number for some n), then a closed-form representation of its behavior can be expressed using polynomials and exponentials over rational numbers. We give polytime algorithms for testing whether a loop has periodic rational eigenvalues and determining its closed-form representation. Our algorithms are straightforward to implement, and make no use of algebraic numbers (§5).
   - We identify special cases in which our algorithm can be used to compute closed forms in polynomial arithmetic and linear arithmetic (§5.2). In the linear-arithmetic case, our result coincides with that of Boigelot [2003]; however, our method is polytime, improving upon Boigelot's exponential-space algorithm.
   - We show how to compute, for any linear loop, a linear loop with periodic rational eigenvalues that best approximates its behavior (§6.1).
   - We extend the results for linear loops to the class of solvable polynomial loops with periodic-rational eigenvalues (§8).

(2) We show that the satisfiability problem for the logical fragment that we use to express closed forms is decidable over the rationals (§7). The result yields decision procedures for safety and termination for a class of linear loops.

(3) We demonstrate that the technique is effective in practice, by using it verify safety properties of a suite of integer programs. Compared to state-of-the-art software model checkers on this suite of benchmarks, our abstract interpreter proves the safety of more assertions and is more consistently performant (§9).

§2 presents some additional examples to provide intuition. §3 provides background needed for understanding the paper's results. §4 defines a logic of closed forms, and formulates the problems that the paper addresses. §10 discusses related work.

2 OVERVIEW

A central theme of this paper is the intuition that it is easier to reason about rational numbers than algebraic numbers. Although there are many powerful techniques for computing with algebraic numbers, basic questions about the behavior of non-linear functions over algebraic numbers remain open [Ouaknine and Worrell 2015], and reasoning about algebraic numbers incurs a substantial implementation overhead.

Functions involving exponentials and polynomials over rational numbers are well-behaved in comparison: rational numbers are totally ordered, and this order yields insight into the asymptotic behavior of exponential-polynomial terms—large exponential bases dominate smaller ones, and high-degree polynomials dominate low-degree polynomials (these properties also hold for exponential-polynomials over algebraic reals; see §7.1). This fact, along with quantifier-elimination techniques, allows us to obtain a decidability result for a logic with exponentials and polynomials over the rationals and reals (§7 and §8.1).

This decidability result motivates the question of when it is possible to express the behavior of a loop using only rational numbers. This question is tied to the nature of the eigenvalues of linear transformations. For example, the eigenvalues of Fig. 1(a) are rational (1 and 3), and so the loop admits a closed-form representation over the rationals. The eigenvalues of loop (b) are non-rational

```
while (*) do
    int $a_{\text{old}} = a$;
    int $f_{\text{old}} = f$;
    $a = b + d$;
    $b = c - b - a_{\text{old}}$;
    $f = -c - d - e - f_{\text{old}}$;
    $c = d$;
    $d = e$;
    $e = f_{\text{old}}$;
```

$$\begin{bmatrix} a' \\ b' \\ c' \\ d' \\ e' \\ f' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

Fig. 3. Loop with periodic behavior, and its associated transition matrix.

(i and $-i$), which means that symbolic matrix exponentiation gives a closed-form representation involving non-rational numbers (Eqn. (1)). However, the use of non-rational numbers is not essential because the matrix representing the execution of the loop twice, $\left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)^2 = \left(\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, has only rational eigenvalues (i.e., -1 with multiplicity 2). The squared matrix captures the periodic nature of the loop in Fig. 1(b), enabling us to capture the behavior of the loop with a formula over the rationals by case-splitting on whether the loop iteration is even or odd.

Thus, we can capture the behavior of a linear loop **while** $(*)$ **do** $\{\mathbf{x} = A\mathbf{x}\}$ using rational arithmetic as long as some power A^p of A has all rational eigenvalues. This observation raises the question of how high the power p may be required to make all eigenvalues of a matrix A rational. We prove a bound on the power (as a corollary to Lem. 5.3), but as illustrated by the following example, p may be exponential in the size of A.

Example 2.1. Consider the loop and corresponding transition matrix shown in Fig. 3. The matrix has six distinct eigenvalues, none of which are rational. However, the matrix raised to the 15th power is the 6×6 identity matrix. Following the pattern of Fig. 1(b), one can create a disjunction with 15 cases, as follows:

$$\begin{aligned} &((k \equiv 0 \bmod 15) \wedge (a' - c' = a - c) \wedge (b' = b) \wedge (c' = c) \wedge (d' = d) \wedge (e' = e) \wedge (f' = f)) \\ \vee\ & \cdots \\ \vee\ & \left(\begin{aligned} &(k \equiv 14 \bmod 15) \wedge (a' - c' = -(a - c) - b) \wedge (b' = a - c) \wedge (c' = -c - d - e - f) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge (d' = c) \wedge (e' = d) \wedge (f' = e) \end{aligned} \right) \end{aligned} \tag{4}$$

We observe that although the total period of the loop is 15, its behavior can be decomposed into two smaller periods, 3 and 5. This idea leads to the following more compact formula that also summarizes the behavior of the loop in Fig. 3:

$$\begin{aligned} &\left(\begin{aligned} &\ (k \equiv 0 \bmod 5 \wedge c' = c \wedge d' = d \wedge e' = e \wedge f' = f) \\ &\vee\ (k \equiv 1 \bmod 5 \wedge c' = d \wedge d' = e \wedge e' = f \wedge f' = -c - d - e - f) \\ &\vee\ (k \equiv 2 \bmod 5 \wedge c' = e \wedge d' = f \wedge e' = -c - d - e - f \wedge f' = c) \\ &\vee\ (k \equiv 3 \bmod 5 \wedge c' = f \wedge d' = -c - d - e - f \wedge e' = c \wedge f' = d) \\ &\vee\ (k \equiv 4 \bmod 5 \wedge c' = -c - d - e - f \wedge d' = c \wedge e' = d \wedge f' = e) \end{aligned} \right) \\ \wedge &\left(\begin{aligned} &\ (k \equiv 0 \bmod 3 \wedge a' = c' + a - c \wedge b' = b) \\ &\vee\ (k \equiv 1 \bmod 3 \wedge a' = c' + b \wedge b' = -(a - c) - b) \\ &\vee\ (k \equiv 2 \bmod 3 \wedge a' = c' - (a - c) - b \wedge b' = a - c) \end{aligned} \right) \end{aligned} \tag{5}$$

□

Ex. 2.1 motivates the *periodic rational spectral decomposition* (§5.1), a device that enables the description of the behavior of a loop in terms of its component periods. The periodic rational spectral decomposition makes the description of a loop *additive* rather than *multiplicative* in the factors of its component periods, yielding a polynomial-time algorithm for computing a closed-form representation of the behavior of a loop with periodic rational behavior.

Finally, we may ask how these results may be applied to real programs, which do not simply implement linear transformations. The work of Kincaid et al. [2018, §5] shows how to approximate loops by linear transformations, but these linear transformations may not fall into the class that can be defined using exponentials and polynomials over rationals. The periodic rational spectral decomposition provides an answer to this question as well: we can obtain a *best approximation* of a linear transformation as a linear transformation that can be described in exponential-polynomial rational arithmetic (§6).

3 BACKGROUND

We begin by reviewing some basic facts and notations for polynomials, matrices, and linear maps.

We use $\mathbb{Q}$ to denote the field of rationals. For a field K, we use $K[x_1, ..., x_n]$ to denote the ring of polynomials over the variables $x_1, ..., x_n$ with coefficients in K. A univariate polynomial $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0 \in K[x]$ is said to be **monic** if $a_n = 1$. An **algebraic number** is a complex number that is a root of some polynomial in $\mathbb{Q}[x]$. We use $\overline{\mathbb{Q}}$ to denote the field of algebraic numbers, and $|a + bi| \overset{\text{def}}{=} \sqrt{a^2 + b^2}$ to denote the norm of an algebraic number. Any univariate polynomial $p \in \mathbb{Q}[x]$ of degree n splits into n linear factors over $\overline{\mathbb{Q}}$: $p = (x - \alpha_1) \cdots (x - \alpha_n)$ for some $\alpha_1, ..., \alpha_n \in \overline{\mathbb{Q}}$ (not necessarily all distinct). Each algebraic number $\alpha \in \overline{\mathbb{Q}}$ is associated with a unique **minimal polynomial** $\mu_\alpha \in \mathbb{Q}[x]$, which is the monic polynomial of least degree such that $\mu_\alpha(\alpha) = 0$. For any univariate polynomial $p \in \mathbb{Q}[x]$ and any algebraic number $\alpha \in \overline{\mathbb{Q}}$ such that $p(\alpha) = 0$, we have that μ_α divides p (i.e., there is some $q \in \mathbb{Q}[x]$ such that $p = q\mu_\alpha$).

We use $\mathbb{Q}[k, (-)^k]$ to denote the ring of **exponential polynomials** in a (single) variable k with coefficients in $\mathbb{Q}$:

$$e, e_1, e_2 \in \mathbb{Q}[k, (-)^k] ::= \lambda \mid k \mid \lambda^k \mid e_1 e_2 \mid e_1 + e_2 \qquad \text{where } \lambda \in \mathbb{Q}$$

Similarly, we use $\overline{\mathbb{Q}}[k, (-)^k]$ to denote the ring of exponential-polynomials in a variable k with coefficients in $\overline{\mathbb{Q}}$.

Let $A \in \mathbb{Q}^{n \times n}$ be a square matrix with rational entries. For any $\lambda \in \overline{\mathbb{Q}}$ and any $\mathbf{v} \in \overline{\mathbb{Q}}^n$ such that $\mathbf{v}^T A = \lambda \mathbf{v}^T$ (using $\mathbf{v}^T$ denote the row vector obtained by transposing $\mathbf{v}$), we say that $\mathbf{v}$ is a (left) **eigenvector** of A and λ is an **eigenvalue** of A. A **rank-r generalized (left) eigenvector** of A is a vector $\mathbf{v}$ such that $\mathbf{v}^T(A - \lambda I)^r = 0$ (in particular, rank-1 generalized eigenvectors are exactly eigenvectors). The **generalized eigenspace** of λ is the vector space spanned by the generalized eigenvectors of λ. The **characteristic polynomial** of a matrix $A \in \mathbb{Q}^{n \times n}$ is defined to be $p_A(x) \overset{\text{def}}{=} det(xI - A)$; it is a monic polynomial of degree n whose roots are exactly the eigenvalues of A. The **algebraic multiplicity** of an algebraic number $\lambda \in \overline{\mathbb{Q}}$ is the number of times $(x - \lambda)$ divides p_A; its **geometric multiplicity** is the dimension of the vector space of eigenvectors of λ.

Let $n \in \mathbb{N}$ be a natural number. The body of a (deterministic) numerical loop with n variables can be understood as a function $f : \mathbb{Q}^n \to \mathbb{Q}^n$. We say that f is **linear** if there exists some matrix $A \in \mathbb{Q}^{n \times n}$ such that $f(\mathbf{x}) = A\mathbf{x}$; f is **affine** if there exists $A \in \mathbb{Q}^{n \times n}$ and $\mathbf{b} \in \mathbb{Q}^n$ such that $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$. The behavior of an affine map $f : \mathbb{Q}^n \to \mathbb{Q}^n$ can be understood by analyzing the behavior of the linear map $g(\mathbf{y}) = \begin{bmatrix} A & \mathbf{b} \\ 0 & 1 \end{bmatrix} \mathbf{y}$ on the subspace of $\mathbb{Q}^{n+1}$ in which the last coordinate is 1. Note that in converting from the affine case to the linear case, the algebraic multiplicity of 1 increases by one; in the remainder of the paper, we present results in terms of linear maps unless the result is not robust under such a change. For any $i \in \{1, ..., n\}$, we use $f_i : \mathbb{Q}^n \to \mathbb{Q}$ to denote the map f projected onto the i^{th} coordinate.

Let $f : \mathbb{Q}^n \to \mathbb{Q}^n$ be a function. Define $f^{(-)} : \mathbb{N} \to (\mathbb{Q}^n \to \mathbb{Q}^n)$ to be a function that maps each natural number $k \in \mathbb{N}$ to the k-fold composition of f:

$$f^{(k)} \overset{\text{def}}{=} \underbrace{f \circ \cdots \circ f}_{k \text{ times}} .$$

That is, if f is a function representing the behavior of a loop, then $f^{(k)}$ is function representing the behavior of iterating that loop.

Note that if $f(\mathbf{x}) = A\mathbf{x}$ is a linear function, then $f^{(k)}(\mathbf{x}) = A^k\mathbf{x}$. Thus, describing the iterated behavior of a linear or affine map reduces to describing matrix exponentiation symbolically. A useful tool for describing matrix exponentiation is the **Jordan normal form**. Every matrix A has a Jordan normal form $A = PJP^{-1}$, where J is almost diagonal. More specifically, J is a block-diagonal matrix, where each block along the diagonal is a **Jordan block**. Each Jordan block of A has some eigenvalue of A as its diagonal elements, ones on the superdiagonal, and zeros everywhere else. The algebraic multiplicity of the eigenvalue determines the size of the Jordan block. The geometric multiplicity of an eigenvalue determines the number of Jordan blocks with that eigenvalue on the diagonal.

Our interest in Jordan normal form stems from the fact that a matrix A in Jordan normal form can easily be exponentiated symbolically: $A^k = PJ^kP^{-1}$. For example, let A be a 5×5 matrix with two eigenvalues, λ_1 and λ_2, of geometric multiplicity 1. Suppose that the algebraic multiplicity of λ_1 is 3, and the algebraic multiplicity of λ_2 is 2. We have $A^k = PJ^kP^{-1}$, where

$$J^k = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2 & 1 \\ 0 & 0 & 0 & 0 & \lambda_2 \end{bmatrix}^k = \begin{bmatrix} \lambda_1^k & \binom{k}{1}\lambda_1^{k-1} & \binom{k}{2}\lambda_1^{k-2} & 0 & 0 \\ 0 & \lambda_1^k & \binom{k}{1}\lambda_1^{k-1} & 0 & 0 \\ 0 & 0 & \lambda_1^k & 0 & 0 \\ 0 & 0 & 0 & \lambda_2^k & \binom{k}{1}\lambda_2^{k-1} \\ 0 & 0 & 0 & 0 & \lambda_2^k \end{bmatrix}$$

Given a block-diagonal matrix of Jordan blocks, $J \in K^{n \times n}$, and variable symbol k, we use $exp(J, k)$ to denote the matrix with exponential-polynomial entries such that for any natural number $c \geq n$, we have $exp(J, k)(c) = J^c$, where $exp(J, k)(c)$ denotes the matrix obtained by evaluating each exponential-polynomial entry of $exp(J, k)$ at c.

The ability to exponentiate symbolically is useful for characterizing an iterated linear map, which we illustrate using the transition matrix $A = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ from Fig. 1(b). A's eigenvalues are i and $-i$. Consequently, $f^{(k)}(\mathbf{x}) \overset{\text{def}}{=} A^k\mathbf{x}$ equals

$$\begin{pmatrix} f_1^{(k)}(\mathbf{x_0}) \\ f_2^{(k)}(\mathbf{x_0}) \end{pmatrix} = \begin{pmatrix} i & -i \\ 1 & 1 \end{pmatrix} \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}^k \begin{pmatrix} \frac{-i}{2} & \frac{1}{2} \\ \frac{i}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} \frac{i^k}{2} + \frac{(-i)^k}{2} & \frac{-i*i^k}{2} + \frac{i*(-i)^k}{2} \\ \frac{i*i^k}{2} - \frac{i*(-i)^k}{2} & \frac{i^k}{2} + \frac{(-i)^k}{2} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}.$$

from which one obtains the formula in Eqn. (1).

This example is an illustration of the following well-known fact about the coefficient functions of an iterated linear map:

Theorem 3.1. *Let $f(\mathbf{x}) = A\mathbf{x}$ be a linear map. Let $\lambda_1, ..., \lambda_m$ be the eigenvalues of A. Then for each i, there exist vectors $\mathbf{p}_1(k), ..., \mathbf{p}_m(k) \in \overline{\mathbb{Q}}[k]^n$ of polynomials with algebraic coefficients such that*

$$f_i^{(k)}(\mathbf{x}) = \lambda_1^k(\mathbf{p}_1(k) \cdot \mathbf{x}) + \cdots + \lambda_m^k(\mathbf{p}_m(k) \cdot \mathbf{x}) \tag{6}$$

for all $k \geq n$. Moreover:

- *If each eigenvalue of A is* rational, *then each $\mathbf{p}_i(k)$ has rational coefficients.*
- *If each eigenvalue of A is either 0 or 1, then $f_i^{(k)}(\mathbf{x})$ is a polynomial.*

- *If each eigenvalue of A is either 0 or 1 and A is diagonalizable, then $f_i^{(k)}(\mathbf{x})$ is a linear function.*

4 PROBLEM STATEMENT

We first define the language *EPRA* of **exponential-polynomial rational arithmetic** formulas, which we use to represent the behaviors of numerical loops. Let k denote a distinguished variable symbol (intuitively, the iteration count of a loop). The syntax of *EPRA* is as follows:

$$\begin{aligned}
\lambda &\in \mathbb{Q}\\
m, n &\in \mathbb{N}\\
x &\in \mathsf{Var} = \{x_1, \dots, x_n\}\\
s, t \in \mathsf{Term} &::= \lambda \mid k \mid x \mid \lambda^k \mid st \mid s + t\\
\phi, \psi \in \mathsf{Formula} &::= s < t \mid s \leq t \mid s = t \mid k \equiv m \mod n\\
&\mid \phi \vee \psi \mid \phi \wedge \psi \mid \neg\phi
\end{aligned}$$

Observe that Term is equal to $(\mathbb{Q}[k, (-)^k])[x_1, \dots, x_n]$—the set of polynomials over the variables $x_1, \dots, x_n$ with coefficients drawn from the ring $\mathbb{Q}[k, (-)^k]$ of exponential-polynomials in k. We say that a term t is **linear over** $\mathbb{Q}[k, (-)^k]$ if it can be written as a linear term with coefficients in $\mathbb{Q}[k, (-)^k]$; that is, $t = e_1x_1 + \dots + e_nx_n$, where each $e_i \in \mathbb{Q}[k, (-)^k]$. Similarly, we say that a term is linear over $\mathbb{Q}[k]$ if it can be written as a linear term with coefficients in $\mathbb{Q}[k]$. We say that a term is linear over $\mathbb{Q}$ (or $\mathbb{Z}$) if it can be written as $t = c_0k + c_1x_1 + \dots + c_nx_n$ with each $c_i \in \mathbb{Q}$ (or $\mathbb{Z}$); note that such terms may involve k. We say that a *formula* is linear over $\mathbb{Q}[k, (-)^k]$ (or $\mathbb{Q}[k]$, $\mathbb{Q}$, or $\mathbb{Z}$) if all terms in the formula are linear over $\mathbb{Q}[k, (-)^k]$ ($\mathbb{Q}[k]$, $\mathbb{Q}$, $\mathbb{Z}$, respectively). Note that the formulas that are linear over $\mathbb{Q}$ (or equivalently, $\mathbb{Z}$) are exactly the formulas that are in linear integer arithmetic. We use *EPRA^{lin}*, *PRA^{lin}*, and *LRA* to denote the fragments of *EPRA* that use terms that are linear over $\mathbb{Q}[k, (-)^k]$, $\mathbb{Q}[k]$, and $\mathbb{Q}$, respectively.

We can extend the syntax of *EPRA* to admit terms of the form $\lfloor k/n \rfloor$ and $\lambda^{\lfloor k/n \rfloor}$ (for $n \in \mathbb{N}$). We use a + superscript to denote the extension (e.g., *EPRA$^+$* is *EPRA* extended with such terms). Note that $\lambda^{\lfloor k/n \rfloor}$ denotes a function of sort $\mathbb{N} \to \mathbb{Q}$ (in contrast to $\lambda^{k/n}$, which is not rational-valued). The extension does not change the expressive power of the logic (in a sense formalized in the following lemma)—our interest in the extension is due to the fact that it allows formulas to be more succinct, which we will take advantage of in §5.

Lemma 4.1. *There is an effective procedure to compute from any formula $\phi \in$ EPRA$^+$, a formula $\psi \in$ EPRA that is satisfiable if and only if ϕ is satisfiable. Moreover, $(\exists k \in \mathbb{N}.\phi)$ and $(\exists k \in \mathbb{N}.\psi)$ are equivalent.*

Proof. Let ϕ be an *EPRA$^+$*formula, and let $\underline{n}$ be the least common multiple of all n such that $\lfloor k/n \rfloor$ or $\lambda^{\lfloor k/n \rfloor}$ appears in ϕ. Take $\psi = \bigvee_{r=0}^{\underline{n}-1} \phi[k \mapsto \underline{n}k + r]$ (where $\phi[k \mapsto \underline{n}k + r]$ denotes the formula ϕ with the term $\underline{n}k + r$ substituted for k). Observe that for every term $\lambda^{\lfloor k/n \rfloor}$ in ϕ, $\lambda^{\lfloor k/n \rfloor}[k \mapsto \underline{n}k + r]$ simplifies to $\lambda^{\lfloor r/n \rfloor}(\lambda^{\underline{n}/n})^k$ (with $\lambda^{\lfloor r/n \rfloor}$ and $\lambda^{\underline{n}/n}$ both rational numbers). For every divisibility predicate $k \equiv m \mod n$ in ϕ, we let $d = \gcd(n, \underline{n})$, and let q_n and $q_{\underline{n}}$ be such that $n = dq_n$ and $\underline{n} = dq_{\underline{n}}$—if d divides $r + m$, then $(k \equiv m \mod n)[k \mapsto \underline{n}k + r]$ simplifies to $k \equiv z(r + m)/d \mod q_n$, where z is the multiplicative inverse of $q_{\underline{n}}$ modulo q_n; if d fails to divide $r + m$, then $(k \equiv m \mod n)[k \mapsto \underline{n}k + r]$ simplifies to *false*. □

Definition 4.2. A function $f^{(-)} : \mathbb{N} \to (\mathbb{Q}^n \to \mathbb{Q}^n)$ is **definable** in a language $\mathcal{L}$ (e.g., *EPRA*, *PRA*, ...) if there exists a formula $\phi \in \mathcal{L}$ in $2n + 1$ free variables such that for all $k \in \mathbb{N}$, $\mathbf{x} \in \mathbb{Q}^n$, and $\mathbf{y} \in \mathbb{Q}^n$, we have $\phi(k, \mathbf{x}, \mathbf{y})$ if and only if $f^{(k)}(\mathbf{x}) = \mathbf{y}$. If this holds, we say that the formula ϕ **defines** f.

In general, what one obtains via Eqn. (6) is difficult for a verification tool to work with because of the presence of complex numbers. Boigelot investigated the use of weaker logics to express the closed form of a linear loop, and obtained the following result:

Theorem 4.3 ([Boigelot 2003]). *Let $f(\mathbf{x}) = A\mathbf{x} + b$ be an affine function. If there exists some $p \geq 1$ such that A^p is diagonalizable and all of its eigenvalues are either 0 or 1, then $f^{(-)}$ is definable in linear arithmetic.*

In this paper, our interest lies in the gap between Thm. 3.1 and Thm. 4.3. The primary goal of the work is as follows:

> Given the transition matrix M for a linear loop, find a succinct formula—in a decidable logic—that defines the iterated behavior of M (in the sense of Defn. 4.2).

Toward this end, a secondary goal is to establish that *EPRA* is decidable (see §7).

5 LINEAR LOOPS

This section describes a method for computing succinct formulas that define (in the sense of Defn. 4.2) the behavior of linear loops. §5.1 describes a procedure to compute an $EPRA^{lin+}$ formula that defines the iteration of a linear map that meets certain conditions. §5.2 extends the result of Boigelot [2003], and shows how the algorithms of this section produce representations in even weaker logics in certain cases.

5.1 Logical Exponential-Polynomial Closed Forms

We begin by formalizing the class of matrices in which we are interested, based on properties of their eigenvalues.

Definition 5.1. Let $\lambda \in \overline{\mathbb{Q}}$ be an algebraic number. We say that λ is a **periodic rational** if $\lambda^p \in \mathbb{Q}$ for some $p \in \mathbb{N}$ with $p > 0$. If λ is a periodic rational, we define its **rational period** to be the least $p > 0$ such that $\lambda^p \in \mathbb{Q}$.

Periodic rationals are precisely the roots of polynomials of the form $bx^p - a$, where a and b are integers. Examples include i, $\sqrt[3]{2}$, and $i\sqrt[3]{2}$ which have rational periods of 2, 3, and 6 respectively.

In the remainder of this sub-section, we prove the following result:

Proposition 5.2. *Let $f(\mathbf{x}) = A\mathbf{x}$ be a linear map. There is a polytime algorithm for determining whether each eigenvalue of A is a periodic rational, and if so, computing an $EPRA^{lin+}$ formula that defines $f^{(-)}$.*

As a first step towards Prop. 5.2, we would like to show that it is possible to test whether a given matrix has periodic rational eigenvalues. Given a matrix A with periodic rational eigenvalues, we can enumerate powers $A^1, A^2, A^3, \ldots$ until we find a power A^p with all rational eigenvalues (p is the least common multiple of the rational periods of the eigenvalues of A), but if A does *not* have all rational eigenvalues, this process would go on forever. The following lemma is sufficient to show that there is an upper bound on the powers of A that we need to test to reveal its periodic rational eigenvalues.

Lemma 5.3. *Let $A \in \mathbb{Q}^{n \times n}$. If λ is a periodic rational eigenvalue of A with rational period k, then $k \leq n^3$.*

Proof. Let $p_A(x)$ be A's characteristic polynomial, and let μ_λ be the minimal polynomial of λ. Since λ is a periodic rational, there is some $k \in \mathbb{N}$, and $a, b \in \mathbb{Z}$ such that $\lambda^k = \frac{a}{b}$, and thus λ is a root of the polynomial $bx^k - a$. It follows that μ_λ divides $bx^k - a$ and every root of μ_λ is a root of $bx^k - a$. Since the roots of $bx^k - a$ are all of the form $r\zeta$ where $r = |\lambda|$ and ζ is a root of unity, it follows that μ_λ can be written as $(x - r\zeta_1)\cdots(x - r\zeta_m)$ with each ζ_j a root of unity and such that $r\zeta_1 = \lambda$. In the following, we use ζ to denote $\zeta_1 = \frac{\lambda}{r}$.

Let q be the rational period of r and let $m = \deg(\mu_\lambda)$. Since the constant coefficient of $\mu_\lambda = (x - r\zeta_1)\cdots(x - r\zeta_m)$ is $(-1)^m \Pi_{j=1}^m r\zeta_j$ and must be rational, r^m must be rational, and thus q (the rational period of r) divides m and so $q \leq m$. Because λ is a root of p_A, we have that μ_λ divides p_A and so $\deg(\mu_\lambda) \leq \deg(p_A)$. Summarizing, we have $q \leq m = \deg(\mu_\lambda) \leq \deg(p_A) = n$, and thus $q \leq n$.

Let μ_{λ^q} be the minimal polynomial of λ^q, and let p_{A^q} be the characteristic polynomial of A^q. Reasoning as above, λ^q is a root of $bx^{\frac{k}{q}} - a$ and so μ_{λ^q} can be written as $(x - r^q\zeta_1')\cdots(x - r^q\zeta_{m'}')$ with $\zeta_1' = \zeta^q$. Then

$$\mu_{\lambda^q}(r^q x) = (r^q x - r^q\zeta_1')\cdots(r^q x - r^q\zeta_{m'}') = r^{qm'}(x - \zeta_1')\cdots(x - \zeta_{m'}')$$

is a rational polynomial with ζ^q as a root. Since ζ^{qj} is not real for any $j < \frac{k}{q}$, ζ^q is a primitive d^{th} root of unity for some $d \geq \frac{k}{q} \geq \frac{k}{n}$. Distinguish two cases:

- Case $d > 6$. Since ζ^q is a primitive d^{th} root of unity, we have that Φ_d divides $\mu_{\lambda^k}(r^q x)$, where Φ_d is the d^{th} cyclotomic polynomial. Since μ_{λ^q} divides p_{A^q}, we must have $n = \deg(p_{A^q}) \geq \deg(\mu_{\lambda^q}) \geq \deg(\Phi_d)$. Since the degree of Φ_d is at least $\sqrt{d}$ for $d > 6$, we have $n \geq \sqrt{d}$, and thus $n^2 \geq d$. Since $d \geq \frac{k}{n}$, we conclude $k \leq n^3$.
- Case $d \leq 6$. Since $\frac{k}{n} \leq d \leq 6$, we have $k \leq 6n$ and thus $k \leq n^3$, except when $n = 2$. When $n = 2$, then (reasoning as above) we must have $q \in \{1, 2\}$ and $d \in \{1, 2, 3, 4, 6\}$. The case $q = 2$ and $d = 6$ is not possible by the assumption that ζ^{qj} is not real for $j < d$ (if ζ^2 is a primitive 6^{th} root of unity, then ζ^6 must be real). All other cases have $k = qd \leq 8 \leq n^3$. □

As a corollary of this lemma, we see that if $A \in \mathbb{Q}^{n \times n}$ has all periodic rational eigenvalues, then there is some least power p such that A^p has all rational eigenvalues, and p is bounded by the least common multiple of $\{1, ..., n^3\}$. By Thm. 3.1, we can symbolically exponentiate A^p and define the iterated behavior of A via a formula of the form:

$$\phi(k, \mathbf{x}, \mathbf{y}) = \left(\bigvee_{j=1}^{n} k = j \land \mathbf{y} = A^j \mathbf{x} \right) \lor \left(k > n \land \bigvee_{j=0}^{p-1} (k \equiv j \mod p) \land \mathbf{y} = A^j exp(A^p, \lfloor k/p \rfloor)\mathbf{x} \right) .$$

However, this approach takes exponential space: it requires p case distinctions, and in the worst case, p is exponential in the size of A. The essential issue is illustrated by the matrix from Ex. 2.1 (Fig. 3), which we will refer to as A. The eigenvalues of A are all primitive 3rd and 5th roots of unity, which have rational periods of 3 and 5. The least power p such that A^p has all rational eigenvalues is 15. However, as we will see in this section, it is possible to describe the iterated behavior of A by describing the iterated behavior of A^3 and A^5 on their rational eigenvectors *without* having to enumerate the 15 case distinctions needed to describe the iterated behavior of A in terms of A^{15}.

Our strategy for computing a *succinct* formula that defines the iterated behavior of a linear map with periodic rational eigenvalues is based on a novel technical device: the *periodic rational spectral decomposition* (PRSD). In the following, we will present our strategy in three parts: first, we define PRSD and state some of its properties; then, we show how to compute a PRSD of a matrix; and finally, we show how to compute a formula defining the iterated behavior of a map, given a PRSD.

5.1.1 Periodic Rational Spectral Decomposition (PRSD).

A periodic rational spectral decomposition of a matrix A identifies the (generalized) eigenvectors of *powers* of A that correspond to rational eigenvalues. For matrices A such that A^p has all rational eigenvalues, the PRSD serves a similar role to the Jordan normal form of $A^p = P^{-1}JP$ (noting that the rows of P^{-1} are generalized eigenvectors of A^p). However, unlike with Jordan normal form, the generalized eigenvectors in a PRSD are not required to synchronize on a single period, which allows for a PRSD to be computed in polytime.

Definition 5.4. Let $A \in \mathbb{Q}^{n\times n}$ be a square rational matrix. A **periodic rational spectral decomposition** of A is a set of triples

$$\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\} \subset \mathbb{N} \times \mathbb{Q} \times \mathbb{Q}^n$$

such that

(1) The set $\{\mathbf{v}_1, ..., \mathbf{v}_m\}$ is linearly independent
(2) For all i, $\mathbf{v}_i$ is a generalized eigenvector of A^{p_i} corresponding to λ_i (i.e., there exists some $r \in \mathbb{N}$ such that $\mathbf{v}_i^T(A^{p_i} - \lambda I)^r = 0$).
(3) The set is maximal in the sense that for any vector $\mathbf{u}$ for which there exists a rational number λ and natural numbers p and r such that $\mathbf{u}^T(A^p - \lambda I)^r = 0$ (i.e, $\mathbf{u}$ is a generalized eigenvector of some power of A corresponding to a rational eigenvalue), $\mathbf{u} \in span(\mathbf{v}_1, ..., \mathbf{v}_m)$.

For example, a PRSD of the matrix from Ex. 2.1 (Fig. 3) is as follows: $\{(3, 1, \mathbf{v}_1), (3, 1, \mathbf{v}_2), (5, 1, \mathbf{v}_3), (5, 1, \mathbf{v}_4), (5, 1, \mathbf{v}_5), (5, 1, \mathbf{v}_6)\}$—the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are (left) eigenvectors of A^3 corresponding to the eigenvalue 1 (i.e., $\mathbf{v}_1 A^3 = \mathbf{v}_1$ and $\mathbf{v}_2 A^3 = \mathbf{v}_2$), and $\mathbf{v}_3$ through $\mathbf{v}_6$ are eigenvectors of A^5 corresponding to the eigenvalue 1.

$$\begin{aligned} \mathbf{v}_1^T &= \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & \mathbf{v}_3^T &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & \mathbf{v}_5^T &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{v}_2^T &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{v}_4^T &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \mathbf{v}_6^T &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

While any matrix has a (possibly empty) periodic rational spectral decomposition, conditions 1 and 3 together imply that if all of the eigenvalues of A are periodic rational, then its PRSD spans $\mathbb{Q}^n$. As a result, describing the iterated behavior of A on each vector in its PRSD is sufficient to describe the iterated behavior of A.

5.1.2 Computing a Periodic Rational Spectral Decomposition.

Proposition 5.5. *Alg. 1 is a polytime algorithm for computing a periodic rational spectral decomposition of a matrix.*

Proof. Let $\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\}$ be the set returned by Alg. 1. Conditions 1 and 2 of Defn. 5.4 hold trivially. We prove condition 3. Let $\mathbf{v} \in \mathbb{Q}^n$, $p, r \in \mathbb{N}$ such that $\mathbf{v}^T(A^p - \lambda I)^r = 0$. We must prove that $\mathbf{v} \in \text{span}(\mathbf{v}_1, ..., \mathbf{v}_m)$.

First, a lemma:

> Lemma 5.6. *Let $A \in \overline{\mathbb{Q}}^{n\times n}$ be a square algebraic matrix, let $k \in \mathbb{N}$, and let λ be an eigenvalue of A^k. The generalized eigenspace of A^k corresponding to λ is exactly the span of the generalized eigenspaces of A corresponding to the eigenvalues α of A such that $\alpha^k = \lambda$.*
>
> Proof. Let $\alpha_1, ..., \alpha_d \in \overline{\mathbb{Q}}$ be the eigenvalues of A, and let U_1,...,U_d be the corresponding generalized eigenspaces of A. Let $\lambda_1, ..., \lambda_e \in \overline{\mathbb{Q}}$ be the eigenvalues of A^k, and let P_1, ..., P_e be the corresponding generalized eigenspaces of A^k. For any i let $\{j_{i,1}, ..., j_{i,g_i}\}$ be the set of indices such that $\alpha_{j_{i,1}}^k = \cdots = \alpha_{j_{i,g_i}}^k = \lambda_i$. We must prove that for all i, $P_i = \text{span}(U_{j_{i,1}}, ..., U_{j_{i,g_i}})$.

First, we prove that P_i must contain U_j for any j such that $\alpha_j^k = \lambda_i$. It is sufficient to prove that for all $\mathbf{u} \in \overline{\mathbb{Q}}^n$, $\alpha \in \overline{\mathbb{Q}}$ and $r \in \mathbb{N}$, $\mathbf{u}^T(A - \alpha I)^r = 0$ implies $\mathbf{u}^T(A^k - \alpha^k I)^r = 0$. We proceed by induction on r.

- Base case $r = 1$. $\mathbf{u}^T(A - \alpha I) = 0$ implies $\mathbf{u}^T A = \alpha\mathbf{u}^T$ and so $\mathbf{u}^T A^k = \alpha^k \mathbf{u}^T$ and $\mathbf{u}^T(A^k - \alpha^k I) = 0$
- Induction step. By the induction hypothesis, we have that for all $\mathbf{v}$, $\mathbf{v}^T(A - \alpha I)^r = 0$ implies $\mathbf{v}^T(A^k - \alpha^k I)^r = 0$. Suppose $\mathbf{u}^T(A - \alpha I)^{r+1} = 0$. By induction on k, we can show that $\mathbf{u}^T A^k = \alpha^k\mathbf{u}^T + \mathbf{z}^T$, where $\mathbf{z}^T(A^k - \alpha^k I)^r = 0$:
  - Base case $k = 1$ — trivial.
  - Inductive step. By the induction hypothesis, $\mathbf{u}^T A^k = \alpha^k\mathbf{u}^T + \mathbf{z}^T$ for some $\mathbf{z}$ such that $\mathbf{z}^T(A^k - \alpha^k I)^r = 0$. Then

$$\begin{aligned}
\mathbf{u}^T A^{k+1} &= \mathbf{u}^T A^k A \\
&= (\alpha^k \mathbf{u}^T + \mathbf{z}^T)A \\
&= \alpha^k \mathbf{u}^T A + \mathbf{z}^T A \\
&= \alpha^k(\mathbf{u}^T A + \alpha\mathbf{u}^T - \alpha\mathbf{u}^T) + \mathbf{z}^T A \\
&= \alpha^k(\alpha\mathbf{u}^T + \mathbf{u}^T(A - \alpha I)) + \mathbf{z}^T A \\
&= \alpha^{k+1}\mathbf{u}^T + (\alpha^k\mathbf{u}^T(A - \alpha I) + \mathbf{z}^T A)
\end{aligned}$$

  We now must show that $(\alpha^k\mathbf{u}^T(A - \alpha I) + \mathbf{z}^T A)(A^k - \alpha^k I)^r = 0$. We consider the two parts of the sum separately:
    - ∗ Since $(\mathbf{u}^T(A - \alpha I))(A - \alpha I)^r = 0$, we have by the (outer) induction hypothesis that $(\mathbf{u}^T(A - \alpha I))(A^k - \alpha^k I)^r = 0$, and thus $(\alpha^k\mathbf{u}^T(A - \alpha I))(A^k - \alpha^k I)^r = 0$
    - ∗ Since $\mathbf{z}^T(A^k - \alpha^k I)^r = 0$, we have $\mathbf{z}^T A(A^k - \alpha^k I)^r = \mathbf{z}^T(A^k - \alpha^k I)^r A = 0A = 0$.

  Since $\mathbf{u}^T A^k = \alpha^k\mathbf{u}^T + \mathbf{z}^T$, we have $\mathbf{u}^T(A^k - \alpha^k I) = \mathbf{z}^T$ and so

$$\mathbf{u}^T(A^k - \alpha^k I)^{r+1} = \mathbf{u}^T(A^k - \alpha^k I)(A^k - \alpha^k I)^r = \mathbf{z}^T(A^k - \alpha^k I)^r = 0\ .$$

Since P_i must contain U_j for any j such that $\alpha_j^k = \lambda_i$, we have $P_i \supseteq \text{span}(U_{j_{i,1}}, ..., U_{j_{i,g_i}})$ for all i. Since

$$n = \dim(U_1) + \cdots + \dim(U_d) = \dim(P_1) + \cdots + \dim(P_e)\ ,$$

we have that for all i, P_i is exactly $\text{span}(U_{j_{i,1}}, ..., U_{j_{i,g_i}})$. □

Let $\alpha_1, ..., \alpha_d \in \overline{\mathbb{Q}}$ be the eigenvalues of A such that $\alpha_1^p = \cdots = \alpha_d^p = \lambda$, and let $U_1, ..., U_d$ be the generalized eigenspaces of A corresponding to $\alpha_1, ..., \alpha_d$. From the above lemma, we have that the generalized eigenspace of A^p corresponding to λ is equal to $\text{span}(U_1, ..., U_d)$, and thus $\mathbf{v} \in \text{span}(U_1, ..., U_d)$. Let $p_1, ..., p_d$ be the rational periods of $\alpha_1, ..., \alpha_d$. For each j, U_j belongs to the eigenspace of A^{p_j} corresponding to $\alpha_j^{p_j}$; since (by Lem. 5.3) p_j is bounded above by n^3, we have that U_j is contained in $\text{span}(\mathbf{v}_1, ..., \mathbf{v}_m)$, and therefore $\mathbf{v} \in \text{span}(U_1, ..., U_d) \subseteq \text{span}(\mathbf{v}_1, ..., \mathbf{v}_m)$.

Clearly the number of iterations of each loop is bounded by a polynomial. On line (3), the set of rational eigenvalues of A^p (over which the iteration is performed) can be computed in polytime by computing its characteristic polynomial [Keller-Gehrig 1985] and subsequently factoring it [Lenstra et al. 1982]. □

```
Algorithm 1: PeriodicRationalSpectralDecomposition($A$)
  Data: $A \in \mathbb{Q}^{n \times n}$ a square rational matrix
  Result: Periodic rational spectral decomposition of $A$
  $D \leftarrow \emptyset$;
  for $p \leftarrow 1$ to $n^3$ do
    for each rational eigenvalue $\lambda$ of $A^p$ do
      $B \leftarrow$ basis for the generalized left eigenspace of $A$ corresponding to $\lambda$;
      for $\mathbf{b} \in B$ do
        if $\mathbf{b}$ is not a linear combination of vectors in $D$ then
          $D \leftarrow D \cup \{(p, \lambda, \mathbf{b})\}$,
  return $D$
```

5.1.3 Closed Forms from Periodic Rational Spectral Decompositions. Finally, we can prove Prop. 5.2. Suppose that $f : \mathbb{Q}^n \to \mathbb{Q}^n$ is a linear map with $f(\mathbf{x}) = A\mathbf{x}$. We may use Alg. 1 to compute (in polytime) a periodic rational spectral decomposition $\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\}$. If m is not equal to n, then A has eigenvalues that are not periodic rationals, so we report failure. Otherwise, $m = n$ and $\{\mathbf{v}_1, ..., \mathbf{v}_n\}$ spans $\mathbb{Q}^n$. For any $1 \leq i \leq n$, define $g_i(k, \mathbf{x}) \overset{\text{def}}{=} \mathbf{v}_i^T A^k(\mathbf{x})$ (in the terminology of loops: $g_i(k, \mathbf{x})$ represents the value of the linear term $\mathbf{v}_i^T \mathbf{x}$ as a function of the initial values of the variables $\mathbf{x}$ and the iteration number k). Since $\{\mathbf{v}_1, .., \mathbf{v}_n\}$ spans $\mathbb{Q}^n$, we can compute a formula that defines the iterated map $f^{(k)}$ by computing formulas that define each of the g_i (in the terminology of loops: the value of any variable can be recovered from the values of the linear terms $\mathbf{v}_1^T\mathbf{x}$,...,$\mathbf{v}_n^T\mathbf{x}_n$). Supposing that for each i, $\phi_i(k, \mathbf{x}, y)$ is a formula that defines g_i, then the following formula defines $f^{(k)}$:

$$\phi(k, \mathbf{x}, \mathbf{y}) \overset{\text{def}}{=} \bigwedge_{i=1}^{n} \phi_i(k, \mathbf{x}, \mathbf{v}_i^T \mathbf{y}) \ .$$

We now address how to compute, for a given i, a formula $\phi_i(k, \mathbf{x}, y)$ that defines $g_i(k, \mathbf{x}) = \mathbf{v}_i^T A^k(\mathbf{x})$. By assumption, $\mathbf{v}_i$ is a generalized eigenvector of A^{p_i} corresponding to the eigenvalue λ_i, so there is some r such that $\mathbf{v}_i^T(A^{p_i} - \lambda_i I)^r = 0$. We can compute the least such r by taking

$$\mathbf{u}_1 \overset{\text{def}}{=} \mathbf{v}_i, \mathbf{u}_2 \overset{\text{def}}{=} (A^{p_i} - \lambda I)^T \mathbf{u}_1, \mathbf{u}_3 \overset{\text{def}}{=} (A^{p_i} - \lambda I)^T \mathbf{u}_2, ...$$

until we reach a number r such that $\mathbf{u}_{r+1} = 0$ (the sequence $\mathbf{u}_1, ..., \mathbf{u}_r$ is known as the *Jordan chain* of $\mathbf{v}_i$). Let $U \in \mathbb{Q}^{r \times n}$ be the matrix whose rows are $\mathbf{u}_1, ..., \mathbf{u}_r$. Then the sequence of equations defining the sequence $\mathbf{u}_1, ..., \mathbf{u}_r$ can be rearranged into the equation $UA^{p_i} = JU$, where $J \in \mathbb{Q}^{r \times r}$ is a Jordan block with λ_i on the diagonal. Let $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$. For any $k \in \mathbb{N}$, there exists q and s such that $k = qp_i + s$ and $0 \leq s < p_i$, and we have

$$g_i(k, \mathbf{x}) = \mathbf{v}_i^T A^{qp_i + s} \mathbf{x} = \mathbf{e}_1^T U A^{qp_i + s} \mathbf{x} = \mathbf{e}_1^T J^q U A^s \mathbf{x}$$

It follows that the formula

$$\phi_i(k, \mathbf{x}, y) \overset{\text{def}}{=} \left(\bigvee_{j=0}^{r} k = j \wedge y = \mathbf{v}_i^T A^j \mathbf{x} \right) \vee \left(k > r \wedge \bigvee_{s=0}^{p_i - 1} k \equiv s \mod p_i \wedge y = \mathbf{e}_1^T (exp(J, \lfloor k/p_i \rfloor) U A^s \mathbf{x} \right)$$

is an $EPRA^{lin+}$ formula that defines $g_i(k, \mathbf{x})$.

5.2 Polynomial and Linear Closed Forms

In the preceding sections, we showed that a linear loop can be expressed in exponential-polynomial arithmetic using just rational numbers, provided the transformation matrix has eigenvalues that are all periodic rational. In some applications it may be desirable to express loops in weaker theories, such as polynomial arithmetic (supported by the NIRA theory in SMTLIB) or linear arithmetic. Both cases can be handled using essentially the same technique presented previously in this section.

First, we consider the polynomial arithmetic case. Let $f(\mathbf{x}) = A\mathbf{x}$ be a linear map. Supposing that the eigenvalues of A are either 0 or a root of unity, then §5.1 computes an exponential-polynomial formula that defines $f^{(-)}$. Every sub-term that is an exponential will be of the form 0^k, 1^k, or $(-1)^k$, which can be simplified to 0, 1, and a case split between 1 and -1, respectively. The following result follows:

Corollary 5.7. *Let $f(\mathbf{x}) = A\mathbf{x}$ be a linear map. There is a polytime algorithm for determining whether each eigenvalue of A is a root of unity, and if so, computing a PRA^{lin+} formula that defines $f^{(-)}$.*

The question of when an iterated linear map can be expressed in linear arithmetic was resolved by Boigelot (Thm. 4.3). However, Boigelot's construction of a formula that defines $f^{(k)}$ requires exponential space, because it constructs the power A^p for which A has all eigenvalues in $\{0, 1\}$. By employing the periodic rational spectral decomposition, it is possible to improve on Boigelot's result, and construct a linear arithmetic formula in polytime:

Corollary 5.8. *Let $f(\mathbf{x}) = A\mathbf{x} + b$ be an affine function. There is a polytime algorithm for determining whether there exists some $p \geq 1$ such that A^p is diagonalizable and all of its eigenvalues are either 0 or 1, and if so, computing an LRA formula that defines $f^{(-)}$.*

6 APPROXIMATING GENERAL LOOPS

The last section showed how to obtain closed-form representations of a simple class of loops of the form **while** $(*)$ **do** $\{\mathbf{x} = A\mathbf{x}\}$, where A is a square rational matrix with periodic rational eigenvalues. In this section, we show how these results can be put to practical use in program analysis. In particular, we discuss how to obtain formulas that over-approximate the behavior of linear loops with arbitrary eigenvalues, general loops (with conditional branching, nested loops, etc.) and recursive procedures, and loops with guards.

6.1 Approximating General Linear Maps

§5 showed how to compute closed forms for iterated linear maps that satisfy certain conditions. In this sub-section, we ask: what can we do with a linear map that does *not* satisfy these conditions? We will show that it is possible to compute a *best abstraction* of a linear map that does satisfy these conditions, which can be used to over-approximate the iterated behavior of the original map.

The key idea is the observation that if the procedure outlined in §5.1 is allowed to continue (instead of reporting failure) when given an input matrix that has eigenvalues that are not periodic rationals, it will produce *some* formula $\phi(k, \mathbf{x}, \mathbf{y})$. The formula always over-approximates the behavior of the iterated linear map (and captures its behavior exactly when all the eigenvalues are periodic rationals). The nature of this over-approximation is formalized in the following.

Let $f : \mathbb{Q}^n \to \mathbb{Q}^n$ be a linear map, and let $\mathcal{A}$ be a class of linear maps (e.g., linear maps with rational eigenvalues). A *linear abstraction* of f in $\mathcal{A}$ consists of a pair of functions $\alpha : \mathbb{Q}^n \to \mathbb{Q}^m$ and $f^\sharp : \mathbb{Q}^m \to \mathbb{Q}^m$ such that $\alpha \circ f = f^\sharp \circ \alpha$ and $f^\sharp \in \mathcal{A}$. Any linear abstraction of a function over-approximates its behavior; we are interested in the abstraction that is best (most precise). We say that a linear abstraction $(\alpha, f^\sharp)$ of f in $\mathcal{A}$ is a **best abstraction** if for any other linear abstraction (β, g) of f in $\mathcal{A}$, there is some linear transformation $\overline{\alpha}$ so that $\overline{\alpha} \circ \alpha = \beta$.[1]

[1] In the language of category theory: let **LDS** be the category of linear dynamical systems, where the objects are linear maps from a rational vector space to itself and arrows are *linear simulations*: we have an arrow $\alpha : f \to f^\sharp$ iff $\alpha \circ f = f^\sharp \circ \alpha$. We

Proposition 6.1. *The class of linear maps with rational eigenvalues admits best abstractions.*

Proof. Let $f(\mathbf{x}) = A\mathbf{x}$ be a linear map of dimension n and let $\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\}$ be a periodic rational spectral decomposition of A. Let V be the matrix whose rows are $\mathbf{v}_1^T, ..., \mathbf{v}_m^T$.

First, we show that there exists a unique square matrix $U \in \mathbb{Q}^{m \times m}$ such that $VA = UV$. Let P be the algebraic vector space spanned by the generalized eigenvectors of periodic rational eigenvalues of A:

$$P = span\{\mathbf{v} : \exists \lambda \in \overline{\mathbb{Q}}. \exists p \geq 1. \exists r \geq 1. \lambda^p \in \mathbb{Q} \wedge \mathbf{v}^T(A - \lambda I)^r = 0\} \ .$$

It is easy to check that for all $\mathbf{v} \in P$ we have $\mathbf{v}^T A \in P$. By Lem. 5.6, $\{\mathbf{v}_1, ..., \mathbf{v}_m\}$ is a basis for P, so for all $i \in \{1, ..., m\}$ there exists a unique $\mathbf{u}_i$ such that $\mathbf{v}_i^T A = \mathbf{u}_i^T V$. Taking U to be the matrix whose rows are $\mathbf{u}_1^T, ..., \mathbf{u}_m^T$, we have $VA = UV$.

Next, we show that U has periodic rational eigenvalues. Suppose that $\mathbf{w}$ is a nonzero vector and that $\mathbf{w}^T U = \lambda \mathbf{w}^T$ for some λ. Then $\mathbf{w}^T UV = \lambda \mathbf{w}^T V$, and since $UV = VA$ we have $(\mathbf{w}^T V)A = \lambda(\mathbf{w}^T V)$. It follows that either $\mathbf{w}^T V$ is 0 or $\mathbf{w}^T V$ is a left eigenvector of A with corresponding eigenvalue λ. Since the rows of V are linearly independent, and $\mathbf{w}$ is nonzero, $\mathbf{w}^T V$ is nonzero. Since $\mathbf{w}^T V$ is a left eigenvector of A with corresponding eigenvalue λ and is in P, we have that λ must be periodic rational.

Finally, we show that (V, U) is a *best* abstraction. Suppose that (W, T) is another abstraction of A (i.e., $TW = WA$ and T has periodic rational eigenvalues) with dimension m'. Let $\{(q_1, \alpha_1, \mathbf{s}_1'), ...(q_{m'}, \alpha_{m'}, \mathbf{s}_{m'})\}$ be a PRSD of T, and let S be the matrix whose rows are $\mathbf{s}_1, ..., \mathbf{s}_{m'}$. By assumption the eigenvalues of T are periodic rational, so its generalized eigenvalues span $\mathbb{Q}^{m'}$ and S is invertible. Observe that for every q, r, and α, we have

$$(T^q - \alpha I)^r W = \left(\sum_{i=0}^{r} \binom{r}{i} T^{qi} (-\alpha)^{r-i} \right) W = \sum_{i=0}^{r} \binom{r}{i} W A^{qi} (-\alpha)^{r-i} = W(A^q - \alpha I)^r \ .$$

We construct a matrix Z such that for each row $\mathbf{z}_i^T$ we have $\mathbf{z}_i^T V = \mathbf{s}_i^T W$ as follows. For each $\mathbf{s}_i$, there is some r such that $\mathbf{s}_i^T (T^{q_i} - \alpha_i I)^r = 0$. From the above argument, we have that $\mathbf{s}_i^T W(A^{q_i} - \alpha_i I)^r = \mathbf{s}_i^T (T^{q_i} - \alpha_i I)^r W = 0$, and so $\mathbf{s}_i^T W$ is either 0 or a generalized periodic eigenvector of A. In the former case define $\mathbf{z}_i$ to be 0, and in the latter define $\mathbf{z}_i$ to be the unique solution to $\mathbf{z}_i^T V = \mathbf{s}_i^T W$. We have $ZV = SW$, and so by taking $\overline{W} = S^{-1}Z$ we have $\overline{W}V = S^{-1}ZV = S^{-1}SW = W$. □

Example 6.2. Consider the linear loop

$$\textbf{while } (*) \textbf{ do} \quad \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -1 & -2 & 0 \\ 3 & 1 & 2 & 2 \\ -1 & 0 & 0 & -1 \\ 4 & 1 & 1 & 2 \end{bmatrix}}_{A} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \tag{7}$$

This loop differs from the loops in Fig. 1(a), Fig. 1(b), and Fig. 3, in that its transformation matrix has eigenvalues that are not periodic rational. That is, there is no power p for which the eigenvalues of $A^p_{Eqn.\ (7)}$ are rational. However, the transformation matrix of Eqn. (7) does exhibit *some* periodic rational behavior. In particular, the four eigenvalues of $A^4_{Eqn.\ (7)} = \begin{pmatrix} -4 & 0 & 0 & 0 \\ 66 & -28 & -66 & 24 \\ -33 & 12 & 29 & -12 \\ 78 & -33 & -78 & 29 \end{pmatrix}$ are -4,

say that a subcategory of **LDS** admits best abstractions if the inclusion functor into **LDS** has a left adjoint. See [Kincaid 2018] for more details on this view.

with multiplicity 2, and non-rationals that are approximately 33.9706 and 0.0294373. In other words, for $p = 4$ *some* of the eigenvalues of $A^p_{Eqn.\ (7)}$ are rational, and some are not.

The eigenvalues of $A_{Eqn.\ (7)}$ are $1 \pm i$ and $1 \pm \sqrt{2}$. While $1 + i$ and $1 - i$ are periodic rationals, $1 + \sqrt{2}$ and $1 - \sqrt{2}$ are not. The periodic rational spectral decomposition of $A_{Eqn.\ (7)}$ is $\{(4, -4, \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}), (4, -4, \begin{bmatrix} 0 & 1 & 2 & 0 \end{bmatrix})\}$ from which we see that the best linear abstraction of $A_{Eqn.\ (7)}$ is $\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\right)$, which can be realized as the loop

$$\begin{array}{l} \textbf{while } (*) \textbf{ do} \\ \quad \begin{bmatrix} w \\ x + 2y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} w \\ x + 2y \end{bmatrix} \end{array} \tag{8}$$

This abstraction of Eqn. (7) yields the following over-approximation of Eqn. (7)'s behavior:

$$\begin{aligned} & \left(k \equiv 0 \bmod 4 \land w' = (-4)^{\lfloor \frac{k}{4} \rfloor} w \land (x' + 2y') = (-4)^{\lfloor \frac{k}{4} \rfloor}(x + 2y)\right) \\ \lor\ & \left(k \equiv 1 \bmod 4 \land w' = (-4)^{\lfloor \frac{k}{4} \rfloor}(w - x - 2y) \land (x' + 2y') = (-4)^{\lfloor \frac{k}{4} \rfloor}(w + x - 2y)\right) \\ \lor\ & \left(k \equiv 2 \bmod 4 \land w' = (-4)^{\lfloor \frac{k}{4} \rfloor}(-2x - 4y) \land (x' + 2y') = (-4)^{\lfloor \frac{k}{4} \rfloor}(2w)\right) \\ \lor\ & \left(k \equiv 3 \bmod 4 \land w' = (-4)^{\lfloor \frac{k}{4} \rfloor}(-2w - 2x - 4y) \land (x' + 2y') = (-4)^{\lfloor \frac{k}{4} \rfloor}(2w - 2x + 4y)\right) . \end{aligned} \tag{9}$$

Eqn. (9) expresses an overapproximation of the behavior of Eqn. (7) because Eqn. (8) tracks only the values of variable w and the expression $x + 2y$. Moreover, because Eqn. (8) is the *best linear-loop abstraction* of Eqn. (7), Eqn. (9) is the best closed form for Eqn. (7) that is expressible in exponential-polynomial rational arithmetic. □

6.2 Control Flow and Recursive Procedures

We now discuss how we may analyze the behavior of general programs. First, we consider a simple structured programming language. Let X denote a finite set of program variables, and define the syntax of programs as follows:

$$\begin{aligned} s, t \in \mathsf{Expr} &::= \mathtt{x} \in X \mid n \in \mathbb{Z} \mid s + t \mid st \\ c \in \mathsf{Cond} &::= s \leq t \mid s = t \mid s \leq c_1 \land c_2 \mid c_1 \lor c_2 \mid \neg c \\ P \in \mathsf{Program} &::= x \ \mathtt{:=}\ t \mid P_1 \mathtt{;} P_2 \mid \mathtt{if}\ c\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2 \mid \mathtt{while}\ c\ \mathtt{do}\ P \end{aligned}$$

A *transition formula* is a formula (in the language defined in §4, extended with existential quantification) over the program variables X and a set of primed copies X', representing the values of the program variables before and after executing some program. Our goal is to compute, for any given program P, a transition formula $TF[\![P]\!]$ that over-approximates its behavior. Such a formula can be computed by recursion on the program's syntax:

$$\begin{aligned} TF[\![\mathtt{x\ :=\ e}]\!] &\overset{\text{def}}{=} \mathtt{x}' = e \land \bigwedge_{\mathtt{y} \neq \mathtt{x} \in X} \mathtt{y}' = \mathtt{y} \\ TF[\![\mathtt{if}\ c\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2]\!] &\overset{\text{def}}{=} (c \land TF[\![P_1]\!]) \lor (\neg c \land TF[\![P_2]\!]) \\ TF[\![P_1 \mathtt{;} P_2]\!] &\overset{\text{def}}{=} \exists X''. TF[\![P_1]\!][X' \mapsto X''] \land TF[\![P_2]\!][X \mapsto X''] \\ TF[\![\mathtt{while}\ c\ \mathtt{do}\ P]\!] &\overset{\text{def}}{=} loop(c \land TF[\![P]\!]) \land (\neg c[X \mapsto X']) \end{aligned}$$

where *loop* is a function that over-approximates the transitive closure of a transition formula. Thus, the essential problem is to design the function *loop*.

We now show how to use the results of the previous section to implement a function *loop* that over-approximates transitive closure. Let F be a transition formula. Using the algorithm from [Kincaid et al. 2018, §5.3], we may compute an affine transformation that simulates F, in the following sense. The algorithm computes a (*simulation*) matrix $S \in \mathbb{Q}^{n \times |X|}$, a (*transformation*) matrix $A \in \mathbb{Q}^{n \times n}$, and a vector $\mathbf{b} \in \mathbb{Q}^n$ such that $F(X, X') \models S\mathbf{x}' = A(S\mathbf{x}) + \mathbf{b}$, where $\mathbf{x}$ and $\mathbf{x}'$ are column vectors containing the variables X and X', respectively. The entailment $F(X, X') \models S\mathbf{x}' = A(S\mathbf{x}) + \mathbf{b}$ can be understood as saying that for every transition of the formula F, there is a corresponding transition of the affine map $f(\mathbf{y}) = A\mathbf{y} + \mathbf{b}$, where the correspondence between the state-spaces of F and f is given by the simulation matrix S. We may represent the affine transformation f as a linear transformation by adding a dimension: define

$$\hat{A} \overset{\text{def}}{=} \begin{bmatrix} A & \mathbf{b} \\ 0 & 1 \end{bmatrix} \qquad \hat{S} \overset{\text{def}}{=} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \qquad \hat{\mathbf{x}} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \qquad \hat{\mathbf{x}}' \overset{\text{def}}{=} \begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix}$$

Let $\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\}$ be a periodic rational spectral decomposition of $\hat{A}$. By the results of the previous section, for each $i \in \{1, ..., m\}$, we can compute a formula $\phi_i(k, \mathbf{y}, z)$ such that $\phi_i(k, \mathbf{y}, z)$ holds exactly when $z = \mathbf{v}_i^T \hat{A}^k(\mathbf{y})$. Finally, we take:

$$loop(F) \overset{\text{def}}{=} \exists k \in \mathbb{N}. \bigwedge_{i=1}^{m} \phi_i(k, \hat{S}\hat{\mathbf{x}}, \mathbf{v}_i^T \hat{S}\hat{\mathbf{x}}') \ .$$

Thus we have shown that the techniques introduced in the last section can be used to analyze programs in a simple structured programming language. Following [Farzan and Kincaid 2015], this analysis can be extended to programs with arbitrary control flow (e.g., `goto`) using the framework of algebraic program analysis [Tarjan 1981a,b]. Following [Kincaid et al. 2017], this analysis can extended to a language with recursive procedures (using the same function *loop* to analyze recursion) using a tensor-product construction [Reps et al. 2016].

Example 6.3. Consider loop (a) given below:

```
while (*) do
    int tmp = x + z − y;
    if (*) x = x + y;
    else z = z + y;
    y = tmp;
          (a)
```

```
while (*) do
```

$$\begin{bmatrix} x+z \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x+z \\ y \end{bmatrix} \qquad (10)$$

(b)

We cannot characterize the value-sequences of x and z because of the nondeterministic branch in the loop body; however, we *can* characterize the value-sequence of the sum $x + z$. In particular, the sequence for $x + z$ and y is

$$\begin{pmatrix} x_0 + z_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} x_0 + z_0 + y_0 \\ x_0 + z_0 - y_0 \end{pmatrix}, \begin{pmatrix} 2x_0 + 2z_0 \\ 2y_0 \end{pmatrix}, \begin{pmatrix} 2x_0 + 2z_0 + 2y_0 \\ 2x_0 + 2z_0 - 2y_0 \end{pmatrix}, \begin{pmatrix} 4x_0 + 4z_0 \\ 4y_0 \end{pmatrix}, \begin{pmatrix} 4x_0 + 4z_0 + 4y_0 \\ 4x_0 + 4z_0 - 4y_0 \end{pmatrix}, \dots$$

In essence, we can track the values produced by the alternative, non-branching loop (b). From this loop, we obtain the three-part formula

$$\begin{aligned} & ((x' + z' = x_0 + z_0) \land (y' = y_0) \land (k = 0)) \\ \lor\ & \left((x' + z' = 2^{\lfloor k/2 \rfloor}(x_0 + z_0)) \land (y' = 2^{\lfloor k/2 \rfloor} y_0) \land (k > 0) \land (k \equiv 0 \bmod 2)\right) \\ \lor\ & \begin{pmatrix} (x' + z' = 2^{\lfloor k/2 \rfloor}(x_0 + z_0) + 2^{\lfloor k/2 \rfloor} y_0) \\ \land\ (y' = 2^{\lfloor k/2 \rfloor}(x_0 + z_0) - 2^{\lfloor k/2 \rfloor} y_0) \land (k \equiv 1 \bmod 2) \end{pmatrix} \end{aligned} \qquad (11)$$

6.3 Approximating Loop Guards

The methods that we have developed so far have assumed that loops have nondeterministic guards. However, the guard of a loop is typically crucial to reasoning about its behavior. In this section, we explain how to approximate loop guards.

Given a transition formula $F(X, X')$ representing the action of the loop, we can recover information about the pre-condition of the loop with the formula $\exists X'.F(X, X')$ and we can recover information about the post-condition of the loop with the formula $\exists X.F(X, X')$. We may then strengthen the formula *loop*(F) with the conjunct $k = 0 \lor ((\exists X'.F(X, X')) \land (\exists X.F(X, X')))$.

The strengthened formula ensures that the pre-condition of the loop holds in the initial state (and the post-condition holds in the final state). Ideally, we would like to have a formula that ensures that the pre-condition holds at every intermediate state. As shown by Finkel and Leroux [2002], such a formula can be computed in the case that F is a linear formula and its reachability relation is definable in Presburger arithmetic, by employing quantifier elimination for Presburger arithmetic. Using the periodic rational spectral decomposition, we obtain a formula that is equivalent for loops that satisfy the above condition, and produces an over-approximation for loops that do not.

Let $F(X, X')$ be a transition formula, let $f(\mathbf{y}) = A\mathbf{y} + \mathbf{b}$ be an over-approximating affine map with simulation matrix S, and let $\hat{A}$, $\hat{S}$, $\hat{\mathbf{x}}$, and $\hat{\mathbf{x}}'$ be as above. Let $\{(p_1, \lambda_1, \mathbf{v}_1), ..., (p_m, \lambda_m, \mathbf{v}_m)\}$ be a PRSD of $\hat{A}$. For any i, we say that $\mathbf{v}_i^T \hat{A}\hat{\mathbf{x}}$ is a Presburger-definable term if its dynamics are governed by a Presburger arithmetic formula (i.e., $\phi_i(k, \hat{S}\hat{\mathbf{x}}, \mathbf{v}_i^T \hat{S}\hat{\mathbf{x}}')$ is in Presburger arithmetic, or equivalently $\lambda_i \in \{-1, 0, 1\}$ and $\mathbf{v}_i^T \hat{A}^p = \lambda_i \mathbf{v}$). Let L be the set of indices of Presburger-definable terms:

$$L \overset{\text{def}}{=} \{i \in \{1, ..., m\} : \lambda_i \in \{-1, 0, 1\} \land \mathbf{v}_i^T \hat{A}^p = \lambda_i \mathbf{v}^T\}$$

Let F_{lin} be a linear formula that over-approximates F [Farzan and Kincaid 2015, §IV]. Define a formula P to be the formula F_{lin} projected onto the space spanned by the Presburger-definable terms of F:

$$P \overset{\text{def}}{=} \left(\exists X, X'.\left(F_{lin}(X, X') \land \bigwedge_{i \in L} z_i = \mathbf{v}_i^T \hat{A}\hat{\mathbf{x}}\right)\right)$$

where the z_i's are fresh variables introduced to represent each Presburger-definable term. Define a formula G that constrains the Presburger-definable terms of F to satisfy the guard P at every iteration before k:

$$G \overset{\text{def}}{=} \forall \ell \in \mathbb{N}.\ell < k \Rightarrow \left(\left(\bigwedge_{i \in L} \phi_i(\ell, \hat{S}\hat{\mathbf{x}}, \mathbf{v}_i^T \hat{S}\hat{\mathbf{x}}')\right) \land (P[z_i \mapsto \mathbf{v}_i^T \hat{A}\hat{\mathbf{x}}']_{i \in L})\right) .$$

The formula G is in Presburger arithmetic, and its quantifiers may be eliminated. Finally, we may strengthen *loop*(F) with G:

$$loop(F) \overset{\text{def}}{=} \exists k \in \mathbb{N}.G \land \bigwedge_{i=1}^{m} \phi_i(k, \hat{S}\hat{\mathbf{x}}, \mathbf{v}_i^T \hat{S}\hat{\mathbf{x}}') .$$

Example 6.4. Consider the following loop:

```
while ($i \neq 10 \land x < 100$) do
    $i = i + 1$;
    $x = x + i$;
```

The term i is Presburger-definable, while x is not (i.e., the technique of [Finkel and Leroux 2002] does not apply). Following the construction above, we obtain

$$G = \forall \ell \in \mathbb{N}.\ell < k \Rightarrow i + \ell \neq 10 \equiv i > 10 \vee i + k \leq 10$$
$$loop(F) = \exists k \in \mathbb{N}.G \wedge i' = i + k \wedge x' = i(i+1)/2 \ .$$

As a result, we can prove that if this loop is executed in a state satisfying the precondition $i = 0 \wedge x = 0$, then the loop will take exactly 10 iterations and terminate in a state satisfying $i = 10 \wedge x = 55$. That is, we see that (for this particular example) having exact information about the Presburger-definable term i allows us to recover exact information about the term x that has non-linear dynamics.

7 DECISION PROCEDURES FOR LINEAR LOOPS

This section establishes decision procedures for fragments of the logic *EPRA* defined in §4, and as a consequence, proves decidability of some problems related to program verification. The main technical result of this section is that the logical fragment required to express closed forms of iterated maps with periodic rational eigenvalues is decidable:

Theorem 7.1. *The satisfiability problem for EPRA^{lin} is decidable over the rationals. That is, there is a procedure that, given a formula $\phi(k, \mathbf{x}) \in EPRA^{lin}$ in m free variables $\mathbf{x}$ plus the distinguished variable k, determines whether there exists some $m \in \mathbb{N}$ and $\mathbf{v} \in \mathbb{Q}^n$ such that $\phi(m, \mathbf{v})$ holds.*

From this theorem and the results of last section (Prop. 5.2), the following two corollaries are immediate:

Corollary 7.2. *The following problem is decidable: given linear arithmetic formulas P and Q and a matrix A with periodic rational eigenvalues, determine whether the Hoare triple*

$$\{P\} \ \ \textbf{\textit{while}}\ (*)\textbf{\textit{do}}\ \mathbf{x} := A\mathbf{x} \ \ \{Q\}$$

is valid.

Corollary 7.3. *The following problem is decidable: given a rational vector $\mathbf{x}_0$, a linear arithmetic formula C, and a matrix A with periodic rational eigenvalues, determine whether the program*

$$\textbf{\textit{while}}\ (C)\ \textbf{\textit{do}}\ \mathbf{x} := A\mathbf{x}$$

terminates starting from $\mathbf{x}_0$.

The proof of Thm. 7.1 proceeds in two steps:

(1) we show how to obtain an equi-satisfiable formula in which the only free variable is the distinguished variable k
(2) we show that it is possible to compute a cut-off value N such that testing satisfiability of the original formula can be reduced to testing satisfiability of a Presburger formula and checking all values less than N.

(1) Eliminate variables. Given a formula ϕ and a variable x (not the distinguished variable k), it is possible to compute a quantifier-free formula equivalent to $\exists x \in \mathbb{Q}.\phi$. The method is essentially the same as [Loos and Weispfenning 1993], adapted to the setting of exponential-polynomials. The idea behind virtual substitution-based quantifier elimination is that although existential quantification conceptually corresponds to an infinite disjunction of substitution instances, $(\exists x.\phi) \equiv \bigvee_{t \in \text{Term}} \phi[x \mapsto t]$, it is possible to represent the infinite disjunction with a finite disjunction of *virtual* substitution instances. That is, rather than the infinite set Term of terms, we take the disjunction over a finite set of *virtual terms* that do not belong to the syntax of our language, but nonetheless substitution can be defined.

Let $A(\phi, x)$ denote the set of atomic subformulas of ϕ in which the variable x appears. Without loss of generality, suppose that each atom in ϕ that contains x is written as $ex < s$ or $ex = s$ (x not in s in either case). There are three virtual terms of interest: the quotient (s/e), with e assumed to be positive; the quotient $(s/e) - \epsilon$, offset by an infinitesimal and e assumed to be positive; and ∞. Define the virtual substitution of a virtual term v for x, denoted $[x/\!/v]$, recursively as follows:

$$\begin{aligned}
(\phi \vee \psi)[x/\!/v] &\overset{\text{def}}{=} (\phi[x/\!/v]] \vee \psi[x/\!/v])\\
(\phi \wedge \psi)[x/\!/v] &\overset{\text{def}}{=} (\phi[x/\!/v]] \wedge \psi[x/\!/v])\\
(\tilde{e}x = t)[x/\!/(s/e)] &\overset{\text{def}}{=} (\tilde{e}s = et)\\
(\tilde{e}x < t)[x/\!/(s/e)] &\overset{\text{def}}{=} (\tilde{e}s < et)\\
(\tilde{e}x = t)[x/\!/(s/e - \epsilon)] &\overset{\text{def}}{=} (\tilde{e} = 0 \wedge t = 0)\\
(\tilde{e}x < t)[x/\!/(s/e - \epsilon)] &\overset{\text{def}}{=} (\tilde{e} \leq 0 \wedge \tilde{e}s < et) \vee (0 < \tilde{e} \wedge \tilde{e}s \leq et)\\
(\tilde{e}x = t)[x/\!/\infty] &\overset{\text{def}}{=} (\tilde{e} = 0 \wedge t = 0)\\
(\tilde{e}x < t)[x/\!/\infty] &\overset{\text{def}}{=} ((\tilde{e} = 0 \wedge 0 < t) \vee \hat{e} < 0)\\
\mathit{atom}[x/\!/v] &\overset{\text{def}}{=} \mathit{atom} \text{ for any atom not containing } x
\end{aligned}$$

Suppose that M is a model of ϕ. Then there are three cases:

(1) There is some $ex = s \in A(\phi, x)$ such that $M \models ex = s$ and $[\![e]\!]^M \neq 0$. If $[\![e]\!]^M > 0$, then we must have $M \models \phi[x/\!/(s/e)]$; otherwise, we have $M \models \phi[x/\!/((-s)/(-e))]$.

(2) There is some $ex < s \in A(\phi, x)$ such that $M \models ex < s$ and $[\![e]\!]^M > 0$. Suppose further that $ex < s$ is selected so that $[\![s]\!]^M/[\![e]\!]^M$ is *least* among all other atoms satisfying this property (i.e., if $e'x < s' \in A(\phi, x)$, $M \models e'x < s'$, and $[\![e']\!]^M > 0$, then $[\![s]\!]^M/[\![e]\!]^M \leq [\![s']\!]^M/[\![e']\!]^M$. Then we have $M \models \phi[x/\!/s/e - \epsilon]$.

(3) None of the above cases hold. Then we have $M \models \phi[x/\!/\infty]$.

Thus, we may take

$$\begin{aligned}
\psi \overset{\text{def}}{=} &\left(\bigvee_{(ex=s)\in A(\phi,x)} (e > 0 \wedge \phi[x/\!/(s/e)]) \vee (e < 0 \wedge \phi[x/\!/((-s)/(-e))] \right)\\
&\vee \left(\bigvee_{(ex<s)\in A(\phi,x)} (e > 0 \wedge \phi[x/\!/(s/e - \epsilon)]) \right)\\
&\vee \phi[x/\!/\infty]
\end{aligned}$$

By the above, ψ is equivalent to $\exists x.\phi$.

By applying this procedure to every variable symbol other than the distinguished variable k, we have reduced the problem of deciding satisfiability of an EPRA^{lin} formula to deciding satisfiability of an EPRA^{lin} formula in which the only variable is k.

(2) Bound solutions. We further reduce the problem to the case that each exponential term λ^k has $\lambda > 0$ by observing that if $\lambda < 0$ we have the following equivalence:

$$\psi \equiv (k \equiv 0 \mod 2 \wedge \psi[\lambda^k \mapsto |\lambda|^k]) \vee (k \equiv 1 \mod 2 \wedge \psi[\lambda^k \mapsto -|\lambda|^k])\ .$$

So suppose w.l.o.g. that the exponential terms of ψ have positive base. We further suppose that each atom in ψ is either a divisibility predicate or a comparison written in the form $e(k) \bowtie 0$ (where $\bowtie \in \{=, <\}$).

We will now show that for each comparison atom $e(k) \bowtie 0$ in ψ, there exists some $N(atom) \in \mathbb{N}$ such that either *atom* is true for all $k \geq N(atom)$ ("*atom* is ultimately true") or *atom* is false for all $k \geq N(atom)$ ("*atom* is ultimately false"). Let *Ultimate*(ψ) denote the (Presburger arithmetic) formula obtained by replacing each comparison atom with its ultimate truth value, and leaving the divisibility predicates unchanged. Letting $N(\psi)$ be the maximum among $N(atom)$ for all atoms appearing in ψ, we have that ψ is satisfiable if and only if *Ultimate*(ψ) is satisfiable or $\psi[k \mapsto m]$ holds for some $m \leq N(\psi)$. So provided that $N(atom)$ is computable, decidability of $EPRA^{lin}$ follows.

Non-trivial exponential-polynomial functions are continuous and have finitely many roots. For any comparison atom $e(k) \bowtie 0$, it is sufficient to choose $N(e(k) \bowtie 0)$ to be any upper bound on the roots of $e(k)$ (since thereafter $e(k)$ does not change sign, and the truth value of $e(k) \bowtie 0$ does not change). Alg. 2 gives an algorithm for finding an upper bound on the roots of an exponential-polynomial. The algorithm is not new—e.g., it is a special case of bounding roots of an exponential-polynomial over the algebraic numbers with a dominant exponential term (see, e.g., [Halava et al. 2005])—we present it here for the sake of completeness and because the rational case is simpler and more accessible. The idea behind the algorithm is that the behavior of an exponential polynomial

$$e(k) = a_1\lambda_1^k k^{d_1} + \cdots + a_n\lambda_n^k k^{d_n}$$

is eventually dominated by the term $a_m\lambda_m^k k^{d_m}$ such that (1) λ_m is greatest among all exponential bases and (2) the degree d_i is greatest among all terms with exponential base λ_m. The function $e(k)$ tends to $\pm\infty$, depending on the sign of the coefficient a_i. Suppose that a_i is positive (and $e(k)$ tends to $+\infty$)—the other case is symmetric. Since multiplying a function by an exponential does not change its sign, it is sufficient to bound the roots of $(1/\lambda_m)^k(e(k))$. We have

$$\begin{aligned}(1/\lambda_m)^k(e(k)) &= \sum_{i=1}^{n} a_i(\lambda_i/\lambda_m)^k k^{d_i}\\ &\geq a_m k^{d_m} + \left(\sum_{\substack{i=1\\ a_i<0}}^{n} a_i(\lambda_i/\lambda_m)^k k^{d_i}\right)\\ &= \underbrace{a_m k^{d_m} + \left(\sum_{\substack{i=1\\ a_i<0,\lambda_i=\lambda_m}}^{n} a_i k^{d_i}\right)}_{\text{polynomial, eventually } \geq 1} + \underbrace{\left(\sum_{\substack{i=1\\ a_i<0,\lambda_i\neq\lambda_m}}^{n} a_i(\lambda_i/\lambda_m)^k k^{d_i}\right)}_{\text{tends to 0}}\end{aligned}$$

Let $\hat{e}(k)$ denote the exponential polynomial above and let $p(k)$ denote the polynomial term on the left-hand side of the sum. We have that $p(k)$ tends to ∞ unless $d_m = 1$ (in which case $p(k)$ is the constant 1 polynomial), and each term $a_i(\lambda_i/\lambda_m)^k k^{d_i}$ is negative on the domain $k \in [0, \infty)$ (since $a_i < 0$) and tends to 0 (since $\lambda_i/\lambda_m < 1$). We may bound the roots of $e(k)$ by finding a number B such that $p(k)$ and each $a(\lambda_i/\lambda_m)^k k^{d_i}$ is increasing on the domain $k \in [B, \infty)$, and then subsequently finding a constant N such that $\hat{e}(N)$ is positive: e may have no roots larger than N.

For the polynomial term $p(k)$, we may find a bound B_p such that $p(k)$ is positive and increasing on $k \in [B_p, \infty)$ by bounding the roots of p and its first derivative; e.g., Cauchy's bound gives $B_p = 1 - a_j/a_m$, where a_j is the smallest (negative) coefficient in p (line (7)). For each exponential-polynomial term $a_i(\lambda_i/\lambda_m)^k k^{d_i}$ we can compute a bound B_i as follows. Consider the term $a_i(\lambda_i/\lambda_m)k^{d_i}$ as a sequence $\left(a_i(\lambda_i/\lambda_m)^k k^{d_i}\right)_{k=0}^{\infty}$. The difference between consecutive terms of this sequence is given by the exponential polynomial

$$\begin{aligned} a_i(\lambda_i/\lambda_m)^{k+1}(k+1)^{d_i} - a_i(\lambda_i/\lambda_m)^k k^{d_i} &= a_i(\lambda_i/\lambda_m)^k((\lambda_i/\lambda_m)(k+1)^{d_i} - k^{d_i}) \\ &= a_i(\lambda_i/\lambda_m)^k\left(((\lambda_i/\lambda_m)-1)k^{d_i} + \sum_{j=0}^{d_i-1}\binom{d_i}{j}k^j\right). \end{aligned}$$

Multiplying the consecutive difference by the exponential $(\lambda_m/\lambda_i)^k$ gives a polynomial that has the same sign as the consecutive difference. Again applying Cauchy's bound, we have that the sequence is negative and increasing on $k \in [B_i, \infty)$ where $B_i = 1 + \binom{d_i}{\lfloor d_i/2 \rfloor}/(1-(\lambda_i/\lambda_m))$ (line (9)). Taking B to be the maximum among B_p and all B_i, we have that $\hat{e}(k)$ is increasing on $k \in [B, \infty)$. We may then do a linear search starting from B for a value N such that $\hat{e}(N)$ is positive (lines (10)–(12)).

Algorithm 2: RootBound(e)

```
   Data: $e(k) = a_1\lambda_1^k k^{d_1} + \cdots + a_n\lambda_n k^{d_n}$, each $\lambda_i > 0$
   Result: Upper bound on the set $\{z \in \mathbb{N} : e(z) = 0\}$
   /* $m$ is the dominant term index                                   */
1  $m \leftarrow$ index such that $\lambda_m = \{\lambda_1, ..., \lambda_n\}$ and the degree $d_m$ is maximal;
2  $\hat{e}(k) \leftarrow a_m x^{d_m} + \sum_{i=1,\ sign(a_i)\neq sign(a_m)}^{n} a_i(\lambda_i/\lambda_m)^k k^{d_i}$ ;   /* Sufficient to bound zeros of $\hat{e}$ */
   /* Find interval $[B, \infty)$ on which $\hat{e}$ is increasing              */
3  $B \leftarrow 0$;
4  for $i = 1$ to $n$ do
5    if $a_i$ and $a_m$ have unequal sign then
6      if $\lambda_i = \lambda_m$ then
7        $B \leftarrow \max(B, 1 + |a_i/a_m|)$;
8      else
9        $B \leftarrow \max(B, 1 + \binom{d_i}{\lfloor d_i/2 \rfloor}/(1-(\lambda_i/\lambda_m)))$;
   /* Find $N \geq B$ with $\hat{e}(E)$ has the same sign as $a_m$              */
10 $N \leftarrow B$;
11 while $\hat{e}(N)$ has the same sign as $a_m$ do
12   $N \leftarrow N + 1$;
13 return $N$
```

7.1 Discussion

The key properties of the field of rational numbers that are exploited in our decision procedure are that (1) all field operations are effective, and (2) rationals are totally ordered. The procedure (and Thm. 7.1) extends immediately to the field of real algebraic numbers, and Cors. 7.2 and 7.3 extend to the field of *periodic real algebraic numbers* (algebraic numbers λ such that $\lambda^p \in \mathbb{R}$ for some $p \in \mathbb{N}$ with $p \geq 1$). Although periodic real algebraic numbers generalize periodic rationals, there are several reasons to prefer periodic rationals: (1) rationals are conceptually simpler, (2) rationals impose significantly lower implementation burden, and (3) eigenvalues that are periodic real algebraic but not periodic rational are rare in our experience (see §9).

8 SOLVABLE POLYNOMIAL MAPS

This section generalizes the results of §5 to *solvable polynomial maps*. Solvable polynomial maps, introduced by Rodríguez-Carbonell and Kapur [2004], are polynomial maps that satisfy certain syntactic conditions (Defn. 8.1) that imply that their dynamics can be captured by a linear map of higher dimension. This property makes solvable polynomial maps amenable to analysis using linear techniques [de Oliveira et al. 2016; Rodríguez-Carbonell and Kapur 2004].

Intuitively, a polynomial map $f : \mathbb{Q}^n \to \mathbb{Q}^n$ is solvable if the dimensions $\{1, \dots, n\}$ can be arranged into strata so that dimensions have non-linear dependencies only upon dimensions of lower strata.

Definition 8.1 ([Rodríguez-Carbonell and Kapur 2004]). A function $f : \mathbb{Q}^n \to \mathbb{Q}^n$ is a **solvable polynomial map** if there exists $S_1, \dots, S_m \subseteq \{1, \dots, n\}$ such that $\{S_1, \dots, S_m\}$ is a partition of $\{1, \dots, n\}$ and for all $1 \leq i \leq m$ we have

$$f_{S_i}(\mathbf{x}) = A_i \mathbf{x}_{S_i} + \mathbf{p}_i(\mathbf{x}_{p_1, \dots, p_{i-1}})$$

where $f_{S_i}(\mathbf{x})$ denotes $f(\mathbf{x})$ projected onto the coordinates S_i, $\mathbf{x}_{S_i}$ denotes $\mathbf{x}$ projected onto the coordinates S_i, $A_i \in \mathbb{Q}^{|S_i| \times |S_i|}$, and $\mathbf{p}_i(\mathbf{x}_{S_1, \dots, S_{i-1}})$ is a column vector (of dimension $|S_i|$) of polynomials over the variables x_j with $j \in S_1 \cup \cdots \cup S_{i-1}$. The **eigenvalues of a polynomial map** are defined to be the eigenvalues of $A_1, \dots, A_m$.

The dynamics of solvable polynomial map can be captured by a linear map by introducing new dimensions to represent non-linear terms [de Oliveira et al. 2016], as shown in the following example.

Example 8.2. Consider the map

$$f(w, x, y, z) = (w + y, -w + x + 2y, x - y, z + wy) \ .$$

Observe that f is solvable:

$$f_{1,2,3}(w, x, y) = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 2 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \end{bmatrix} \qquad f_4(w, x, y, z) = z + xy$$

The function f_4 contains a non-linear term xy. The dynamics of the term xy is given by the product of the terms corresponding to x and y:

$$f_2(w, x, y) f_3(w, x, y) = (-w + x + 2y)(x - y) = (-wx + wy + x^2 + xy - 2y^2)$$

Similarly, we can compute the dynamics of each degree-2 monomial in w, x, and y, and thereby linearize the polynomial map f:

$$f^{(k)}(w, x, y, z) = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \mathbf{e}_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 & -4 & 1 & 4 & 4 \\ 0 & 0 & 0 & 0 & -1 & 1 & 3 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 1 \end{bmatrix}^k \begin{bmatrix} w \\ x \\ y \\ z \\ w^2 \\ wx \\ wy \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ are standard basis vectors. □

The representation of a solvable polynomial map by a linear transformation means that our techniques from §5 apply. The gap is that we must show that if the eigenvalues of a solvable polynomial map are periodic rational, then so are the eigenvalues of its associated linear map. This is indeed the case, yielding the following theorem:

Theorem 8.3. *Let $f : \mathbb{Q}^n \to \mathbb{Q}^n$ be a solvable polynomial map. There is an algorithm for determining whether each eigenvalue of f is a periodic rational, and if so, computing an EPRA$^+$ formula that defines $f^{(-)}$.*

Proof. Let $f : \mathbb{Q}^n \to \mathbb{Q}^n$ be a solvable polynomial map. Consider the ring of polynomials $\mathbb{Q}[x_1, ..., x_n]$ as an (infinite dimensional) vector space over $\mathbb{Q}$. Then f (and indeed, any polynomial map) can be extended to be a linear transformation $\hat{f}$ on this space by defining

$$\hat{f}\left(\sum_i a_i x_1^{d_{i,1}} \cdots x_n^{d_{i,n}})\right) = \sum_i a_i f_1(\mathbf{x})^{d_{i,1}} \cdots f_n(\mathbf{x})^{d_{i,1}} \ .$$

The distinguishing feature of *solvable* polynomial maps is that there is a finite-dimensional invariant subspace that contains $x_1, ..., x_n$, so that $\hat{f}$ can be understood as a linear transformation on that finite-dimensional subspace. More precisely, define $M \subseteq \mathbb{Q}[x_1, ..., x_n]$ to be the least set of monomials that contains $x_1, ..., x_n$ and such that if $m \in M$, then M contains all monomials that appear in $\hat{f}(m)$. Since f is solvable, we have that M is finite.

Enumerate M as $m_1 = x_1^{d_{1,i}} \cdots x_n^{d_{1,i}}$, ..., $m_{|M|} = x_m^{d_{|M|,1}} \cdots x_n^{d_{|M|,n}}$, and let $\mathbf{m}$ be the vector of monomials $\begin{bmatrix} m_1 & \cdots & m_{|M|} \end{bmatrix}^T$. Define $B \in \mathbb{Q}^{|M| \times |M|}$ to be the matrix such that row i of B contains the unique vector $\mathbf{b}_i$ such that $f_1(\mathbf{x})^{d_{i,1}} \cdots f_n(\mathbf{x})^{d_{i,n}} = \mathbf{b}_i^T \mathbf{m}$. Let U be the matrix such that row i of U contains the unique vector $\mathbf{u}_i$ such that $x_i = \mathbf{u}_i^T \mathbf{m}$. Then we have that $f^{(k)}(\mathbf{x}) = UB^k \mathbf{m}$. Supposing that all eigenvalues of B are periodic rational, then by Prop. 5.2 there is an *EPRA^{lin+}* formula $\phi(k, \mathbf{z}, \mathbf{z}')$ that defines the iterated linear map $g^{(-)}$ where $g(\mathbf{z}) = B\mathbf{z}$. The formula $\psi(k, \mathbf{x}, \mathbf{y}) \overset{\text{def}}{=} (\exists \mathbf{z}.\mathbf{y} = U\mathbf{z} \land \phi(k, \mathbf{z}, \mathbf{z}'))[\mathbf{z}' \mapsto \mathbf{q}]$ is an *EPRA$^+$* formula (since the existential quantifier can be eliminated) and defines $f^{(-)}$.

It remains only to show that all eigenvalues of B are periodic rationals. Let $A_1, ..., A_m$. and $\mathbf{y}_1, ..., \mathbf{y}_m$ be as in Defn. 8.1. We show that the eigenvalues of B are products of eigenvalues of $A_1, ..., A_m$; since periodic rationals are closed under products we have the result. We now consider $\hat{f}$ as a linear transformation on the vector space of polynomials with algebraic coefficients $\overline{\mathbb{Q}}[x_1, ..., x_n]$. Intuitively, the matrix B is the representation of f with respect to a particular basis for a particular subspace. We may just as well represent f with respect to another basis, which we construct as follows. For $i \in \{1, .., m\}$, let $V = \{\mathbf{v}_1, ..., \mathbf{v}_{|A_i|}\}$ be a basis for $\overline{\mathbb{Q}}^{|A_i|}$ consisting of generalized eigenvectors of A_i, and take P_i to be the set of polynomials (in fact, linear terms) $P_i = \{\mathbf{v}_j^T \mathbf{y}_i : j \in \{1, ..., |A_i|\}$. Let Q be the set of all finite products of polynomials in P, and observe that Q is a basis for $\overline{\mathbb{Q}}[x_1, ..., x_n]$.

Let $p_1, ..., p_n$ be an enumeration of P in order of increasing P_i, and within each group P_i ordered by the rank of the eigenvector (with lower rank appearing earlier in the enumeration). For any i, let λ_i be the eigenvalue associated with p_i. and define a total order $\preceq$ on Q by: $p_1^{d_1} \cdots p_n^{d_n} \preceq p_1^{e_1} \cdots p_n^{e_n}$ iff $d_n \cdots d_1$ is lexicographically less than or equal to $e_n \cdots e_1$. From the fact that each p_i corresponds to a generalized eigenvector of some A_j, we have that $\hat{f}(p_i) = \lambda_i(p_i + p_i')$, where p_i' is a polynomial containing only monomials $\prec p_i$. As a result, for any $q = p_1^{d_1} \cdots p_n^{d_n} \in Q$, we have

$$\begin{aligned}\hat{f}(q) &= \hat{f}(p_1)^{d_1} \cdots \hat{f}(p_n)^{d_n}\\ &= \lambda_1^{d_1} \cdots \lambda_n^{d_n}(p_1 + p_1')^{d_1} \cdots (p_n + p_n')^{d_n}\\ &= \lambda_1^{d_1} \cdots \lambda_n^{d_n} q + q'\end{aligned}$$

where q' is a polynomial consisting only of monomials that are $\prec q$.

Suppose that a polynomial p is an eigenvector of $\hat{f}$ with eigenvalue λ (i.e., $\hat{f}(p) = \lambda p$)—we must show that λ is a product of eigenvalues of $\lambda_1, ..., \lambda_n$. Let $q = p_1^{d_1} \cdots p_n^{d_n}$ be the greatest monomial with non-zero coefficient in p, and let a be its associated coefficient. We have

$$\lambda p = \hat{f}(p) = \hat{f}(aq + (p - aq)) = a\hat{f}(q) + \hat{f}(p - aq) = a\lambda_1^{d_1} \cdots \lambda_n^{d_n} q + z$$

where z is a polynomial containing only monomials $\prec q$. Since the coefficient of q on the left-hand side must be the same as the coefficient of q on the right-hand side, we have $\lambda = \lambda_1^{d_1} \cdots \lambda_n^{d_n}$. □

8.1 Decision Procedures

This section extends the decidability result established in §7 to the full logic *EPRA*, which allows multiplication between variables other than the distinguished variable k. The satisfiability problem for this logic is decidable over the reals.

Theorem 8.4. *The satisfiability problem for closed-form formulas is decidable over the reals. That is, there is a procedure that, given a formula $\phi(k, \mathbf{x})$ in k and n free variables $\mathbf{x}$, determines whether there is some $c \in \mathbb{N}$ and $\mathbf{v} \in \mathbb{R}^n$ such that $\phi(c, \mathbf{v})$ holds.*

Proof. Write ϕ in disjunctive normal form $\phi \equiv ((D_1 \land C_1) \lor ... \lor (D_n \land C_n))$, where each D_i is a conjunction of divisibility atoms and each C_i is a conjunction of comparison atoms. For each C_i we can compute an equivalent (over the reals) formula C' that only involves the distinguished variable k using quantifier elimination for real closed fields [Tarski 1951]: first replace each exponential term λ^k with a fresh variable symbol x_λ, then eliminate all variables except k and the x_λ variables, then replace each x_λ with λ^k. Thus we have $\phi \equiv ((D_1 \land C_1') \lor ... \lor (D_n \land C_n'))$, with the latter formula being an *EPRA*^{lin} formula—by the previous section, the satisfiability problem for such formulas is decidable. □

9 EVALUATION

Our techniques are implemented on top of ICRA [Farzan and Kincaid 2015; Kincaid et al. 2017, 2018], which uses Z3's UFLRA solver [de Moura and Bjørner 2008], and Apron's NewPolka polyhedron domain [Jeannet and Miné 2009]. We use the NTL number-theory library for computing and factoring the characteristic polynomials of matrices [Shoup 2018].

For loops with all rational eigenvalues—commonly arising in our experience—we expect the results of ICRA using its operational calculus-based recurrence solver (OCRS) [Kincaid et al. 2018] to be largely the same as the one based on periodic rational spectral decomposition (PRSD), and so the main experimental question is one of performance. For loops with non-rational eigenvalues, OCRS introduces function symbols that permit some limited reasoning, while PRSD abstracts away non-periodic rational eigenspaces and treats the remaining periodic rational eigenspaces precisely. Thus, the experimental question is how this trade-off affects precision. Last, the technique from §6.3 can improve precision even on loops with rational eigenvalues (and in particular, loops that only require linear invariant generation), so the experimental question is whether this is effective and performant in practice.

		ICRA+OCRS		ICRA+PRSD		ICRA+PRSD+PG		SeaHorn		UltAuto	
		#safe	time	#safe	time	#safe	time	#safe	time	#safe	time
C4B	35	30	25.5	30	**25.3**	**34**	27.3	29	1833.0	24	3132.6
HOLA	46	**39**	45.2	**39**	**44.4**	**39**	49.8	**39**	1129.9	38	1992.9
lit	20	**17**	12.9	**17**	**12.8**	**17**	13.4	8	603.7	1	3923.6
total	101	86	83.6	86	**82.5**	**90**	90.5	76	3566.6	63	9049.1

Fig. 4. Experimental results

	Min	Mean	Median	Max	Timeout
ICRA	0.4	0.9	0.7	4.5	0
ICRA+PRSD	0.4	0.9	0.7	4.5	0
ICRA+PRSD+PG	0.3	0.9	0.7	5.5	0
SeaHorn	0.1	3.0	0.2	220.8	11
UltAuto	2.0	16.7	3.4	284.5	26

Fig. 5. Timing summary statistics over C4B/HOLA/lit; times reported in seconds, with timeouts excluded.

We ran ICRA in three different configurations: OCRS (operational calculus recurrence solver), PRSD (periodic rational spectral decomposition), PRSD+PG (PRSD along with Presburger guards, §6.3). We also compared against two state-of-the-art software model checkers: Ultimate Automizer [Heizmann et al. 2018] from SV-COMP18, based on predicate abstraction; and SeaHorn [Gurfinkel et al. 2015] version 0.1.0, a Horn-clause solver based on property-directed reachability. We compared these programs on a suite of 101 safe programs, including the *C4B* [Carbonneaux et al. 2015] and *HOLA* [Dillig et al. 2013] suites (which exhibit linear or periodic linear behavior) and *lit*, a selection of loops with non-linear behavior collected from the literature. Our experiments were conducted on a machine running Ubuntu 16.04 equipped with an 8-core Intel(R) Core(TM) i7 1.80GHz processor and 8GB memory, with a time-out of 5 minutes. The experimental results are given in Fig. 4.

We observe that there is no significant performance difference or the number of assertions proved between the PRSD and OCRS solvers. Note, however, that although ICRA+OCRS and ICRA+PRSD both prove 17 of the 20 assertions in the *lit* category, they are not the same 17 assertions. In particular, ICRA+OCRS can prove the correctness of two examples from [Terauchi and Aiken 2005] wherein a precise treatment of the Fibonacci function is required to prove secure information flow (which requires non-periodic rational eigenvalues). ICRA+PRSD can prove a run-time bound for an example from [Tiwari 2004] and establish bounding constraints for an example from [Miné et al. 2016] in which a figure is rotated and scaled in two dimensional space (both of which require precise treatment of complex, but periodic rational, eigenvalues). Our results suggest the hypothesis that loops with eigenvalues that are *not* periodic rational are rare in practice (only 2 loops among 101 programs), although our suite is not sufficiently comprehensive to draw a conclusion. A possible explanation is that loops that are written by human programmers typically involve patterns that result in rational or periodic rational eigenvalues: loop counters, quantities that are doubled or halved, and cyclic quantities.

Although the Presburger Guard technique does incur a performance penalty, we find that it does allow an additional 4 benchmarks in the C4B category to be proved correct. Lastly, we observe that all configurations of ICRA are capable invariant generators: they can prove the correctness of more assertions than SeaHorn and Ultimate Automizer (even for examples that do not require non-linear reasoning). The run-times of the ICRA configurations are more consistent than SeaHorn and Ultimate Automizer (see Fig. 5), and its aggregate run-time is two orders of magnitude faster.

10 RELATED WORK

Closed forms for loops. Boigelot [2003] gives necessary and sufficient conditions for an iterated affine map to be definable in Presburger arithmetic, and also Presburger arithmetic extended with a single function V_r mapping each integer z to the greatest power of r that divides z. Boigelot also considers the case that the linear map is equipped with a polyhedral guard (which can restrict the number of times the linear map is iterated), in which case his conditions are necessary but not sufficient. Finkel and Leroux [2002] extends [Boigelot 2003] to guards defined in Presburger arithmetic, and also considers how to analyze multi-path loops by iterating compositions of affine maps. §5 extends this line of work by giving a polytime procedure for the Presburger case, as well as generalizing to logics beyond Presburger arithmetic.

Jeannet et al. [2014] developed a technique for approximating the behavior of iterated linear maps with arbitrary eigenvalues. The technique is based on approximating the iteration of the real Jordan form of the transition matrix by an abstract domain of template polyhedron matrices. The abstraction technique discussed in §6.1 is of a different nature: we aim to capture the exact dynamics of a subsystem.

The transitive closure of difference-bound relations [Bozga et al. 2006; Comon and Jurski 1998] and octagon relations [Bozga et al. 2009] has been shown to be definable in Presburger arithmetic, and computable in polytime [Konečný 2016]. The theory of ultimately periodic relations unifies work on linear systems and difference-bound/octagon relations [Bozga et al. 2010]. Periodic behavior also features in our work, but the class of matrices we consider (those with periodic rational eigenvalues) does not satisfy the conditions of [Bozga et al. 2010].

Decision problems for linear loops. This paper addresses the problem of computing a closed-form representation of the reachability relation of a linear loop in a decidable logic, which can be used to address a variety of decision problems (e.g., Cors. 7.2 and 7.3) that meet our condition of having all periodic rational eigenvalues. Ouaknine and Worrell [2012, 2015] surveys work on decision problems related to linear loops. One such problem is the *orbit problem*, which can be stated as follows: given a matrix A, an initial vector $\mathbf{s}$ and a target vector $\mathbf{t}$, determine if there is some k such that $A^k\mathbf{s} = \mathbf{t}$. Kannan and Lipton [1986] showed that the orbit problem is decidable in polytime. A generalization of orbit is the *polytope-collision problem* in which we ask whether one polytope is reachable from another. Almagor et al. [2017] shows that this problem is decidable in PSPACE for matrices of dimension at most 3. The result in Cor. 7.2 solves both these problems as a special case, but for a restricted class of matrices.

The *uniform-termination problem* for linear loops was proved to be decidable over the reals by Tiwari [2004], over the rationals by Braverman [2006], and over the integers (for diagonalizable matrices) by Ouaknine et al. [2015]. Uniform termination asks whether any exection of the loop may fail to terminate starting from any initial state. The case considered in Cor. 7.3 allows a pre-state to be specified (i.e., we go beyond *uniform* termination), but again our decidability result holds only for matrices with all periodic rational eigenvalues.

Analysis of polynomial maps. §8 extends our work to the class of *solvable* polynomial maps that was studied by de Oliveira et al. [2016]; Rodríguez-Carbonell and Kapur [2004]. Solvable maps are also related to P-solvable loops [Kovács 2008; Kovács and Jebelean 2006; Kovács et al. 2006] in that solvable polynomial maps are exactly the class of maps that are both polynomial and P-solvable. (The notion of P-solvability is more general in that it admits loops that have non-polynomial (but Gosper-summable) assignments.) The class of *extended* P-solvable loops [Humenberger et al. 2017] captures a strictly larger set of polynomial maps. Humenberger et al. [2018] extends (extended) P-solvable loops to the case of multi-path loops (but not arbitrarily nested loops). The focus of all of the aforementioned work is to compute invariant polynomial equalities of a loop; our work aims to give exact characterizations of a loop's behavior in various arithmetics.

REFERENCES

S. Almagor, J. Ouaknine, and J. Worrell. 2017. The Polytope-Collision Problem. In *ICALP*. 24:1–24:14.

B. Boigelot. 2003. On iterating linear transformations over recognizable sets of integers. *Theor. Comp. Sci.* 309, 1 (2003), 413–468.

M. Bozga, C. Gîrlea, and R. Iosif. 2009. Iterating Octagons. In *TACAS*. 337–351.

M. Bozga, R. Iosif, and F. Konečný. 2010. Fast Acceleration of Ultimately Periodic Relations. In *Computer Aided Verification*. 227–242.

M. Bozga, R. Iosif, and Y. Lakhnech. 2006. Flat Parametric Counter Automata. In *Automata, Languages and Programming*. 577–588.

M. Braverman. 2006. Termination of Integer Linear Programs. In *CAV*.

Q. Carbonneaux, J. Hoffmann, and Z. Shao. 2015. Compositional Certified Resource Bounds. In *PLDI*.

H. Comon and Y. Jurski. 1998. Multiple counters automata, safety analysis and presburger arithmetic. In *CAV*. 268–279.

L. de Moura and N. Bjørner. 2008. Z3: An Efficient SMT Solver. In *TACAS*.

S. de Oliveira, S. Bensalem, and V. Prevosto. 2016. Polynomial Invariants by Linear Algebra. In *ATVA*. 479–494.

I. Dillig, T. Dillig, B. Li, and K. McMillan. 2013. Inductive Invariant Generation via Abductive Inference. In *OOPSLA*.

A. Farzan and Z. Kincaid. 2015. Compositional Recurrence Analysis. In *FMCAD*.

A. Finkel and J. Leroux. 2002. How to Compose Presburger-Accelerations: Applications to Broadcast Protocols. In *FST TCS*. 145–156.

A. Gurfinkel, T. Kahsai, A. Komuravelli, and J.A. Navas. 2015. The SeaHorn Verification Framework. In *CAV*.

V. Halava, T. Harju, M. Hirvensalo, and J. Karhumädäki. 2005. *Skolem's Problem – On the Border between Decidability and Undecidability*. Technical Report. Turku Center for Computer Science.

M. Heizmann, Y.-F. Chen, D. Dietsch, M. Greitschus, J. Hoenicke, Y. Li, A. Nutz, B. Musa, C. Schilling, T. Schindler, and A. Podelski. 2018. Ultimate Automizer and the Search for Perfect Interpolants. In *TACAS*. 447–451.

A. Humenberger, M. Jaroschek, and L. Kovacs. 2017. Automated Generation of Non-Linear Loop Invariants Utilizing Hypergeometric Sequences. In *ISSAC*.

A. Humenberger, M. Jaroschek, and L. Kovács. 2018. Invariant Generation for Multi-Path Loops with Polynomial Assignments. In *VMCAI*. 226–246.

B. Jeannet and A. Miné. 2009. Apron: A Library of Numerical Abstract Domains for Static Analysis. In *CAV*.

B. Jeannet, P. Schrammel, and S. Sankaranarayanan. 2014. Abstract Acceleration of General Linear Loops. In *POPL*. 529–540.

R. Kannan and R. J. Lipton. 1986. Polynomial-time Algorithm for the Orbit Problem. *J. ACM* 33, 4 (Aug. 1986), 808–821.

W. Keller-Gehrig. 1985. Fast Algorithms for the Characteristic Polynomial. *Theor. Comput. Sci.* 36, 2-3 (June 1985), 309–317.

Z. Kincaid. 2018. Numerical Invariants via Abstract Machines. In *SAS*.

Z. Kincaid, J. Breck, A. Forouhi Boroujeni, and T. Reps. 2017. Compositional Recurrence Analysis Revisited. In *PLDI*.

Z. Kincaid, J. Cyphert, J. Breck, and T.W. Reps. 2018. Non-Linear Reasoning for Invariant Synthesis. *PACMPL* 2(POPL) (2018), 54:1–54:33.

F. Konečný. 2016. PTIME Computation of Transitive Closures of Octagonal Relations. In *Tools and Algorithms for the Construction and Analysis of Systems*. 645–661.

L. Kovács. 2008. Reasoning Algebraically About P-Solvable Loops. In *TACAS*.

L. Kovács and T. Jebelean. 2006. Finding polynomial invariants for imperative loops in the theorema system. *Proc. of Verify* 6 (2006), 52–67.

L. Kovács, N. Popov, and T. Jebelean. 2006. Combining Logic and Algebraic Techniques for Program Verification in Theorema. In *ISoLA*. IEEE, 67–74.

A. K. Lenstra, H. W. Lenstra, and L. Lovász. 1982. Factoring polynomials with rational coefficients. *Math. Ann.* 261, 4 (1982), 515–534.

R. Loos and V. Weispfenning. 1993. Applying linear quantifier elimination. *The computer journal* 36, 5 (1993), 450–462.

A. Miné, J. Breck, and T. W. Reps. 2016. An Algorithm Inspired by Constraint Solvers to Infer Inductive Invariants in Numeric Programs. In *European Symp. on Programming*. 560–588.

J. Ouaknine, J. Sousa Pinto, and J. Worrell. 2015. On Termination of Integer Linear Loops. In *SODA*. 957–969.

J. Ouaknine and J. Worrell. 2012. Decision Problems for Linear Recurrence Sequences. In *RP*.

J. Ouaknine and J. Worrell. 2015. On Linear Recurrence Sequences and Loop Termination. *ACM SIGLOG News* 2, 2 (April 2015), 4–13.

T. Reps, E. Turetsky, and P. Prabhu. 2016. Newtonian Program Analysis via Tensor Product. In *POPL*.

E. Rodríguez-Carbonell and D. Kapur. 2004. Automatic Generation of Polynomial Loop Invariants: Algebraic Foundations. In *ISSAC*. 266–273.

V. Shoup. 2018. NTL: A library for doing number theory. (2018). http://www.shoup.net/ntl/

R. E. Tarjan. 1981a. Fast Algorithms for Solving Path Problems. *J. ACM* 28, 3 (July 1981), 594–614.

R. E. Tarjan. 1981b. A Unified Approach to Path Problems. *J. ACM* 28, 3 (July 1981), 577–593.

A. Tarski. 1951. A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley, CA. (1951).

T. Terauchi and A. Aiken. 2005. Secure Information Flow as a Safety Problem. In *SAS*. 352–367.

A. Tiwari. 2004. Termination of Linear Programs. In *CAV*.