Reflections on Termination of Linear Loops

Shaowei Zhu and Zachary Kincaid

Princeton University, Princeton NJ 08544, USA
{shaoweiz,zkincaid}@cs.princeton.edu

Abstract. This paper shows how techniques for linear dynamical systems can be used to reason about the behavior of general loops. We present two main results. First, we show that every loop that can be expressed as a transition formula in linear integer arithmetic has a *best* model as a *deterministic affine transition system*. Second, we show that for any linear dynamical system f with integer eigenvalues and any integer arithmetic formula G, there is a linear integer arithmetic formula that holds exactly for the states of f for which G is eventually invariant. Combining the two, we develop a monotone conditional termination analysis for general loops.

1 Introduction

Linear and affine dynamical systems are a model of computation that is easy to analyze (relative to non-linear systems), making them useful across a broad array of applications. In the context of program analysis, affine dynamical systems correspond to loops of the form

$$\textbf{while } (G(\textbf{x})) \textbf{ do } \textbf{x := } A\textbf{x + b} \qquad (\dagger)$$

where G is a formula, A is a matrix, $\mathbf{x}$ is a vector of program variables, and $\mathbf{b}$ is a constant vector. The termination problem for such loops has been shown to be decidable for several variations of this model [29,4,24,9,12]. However, few loops in real programs take this form, and so this work has not yet made an impact on practical termination analysis tools. This paper bridges the gap between theory and practice, showing how techniques for linear and affine dynamical systems can be used to reason about general programs.

Example 1. We illustrate our methodology using the example program in Figure 1 (left). First, observe that although the body of this loop is not of the form ($\dagger$), the value of the sum $x + y$ decreases by z each iteration, and z remains the same. Thus, we can approximate the loop by the linear dynamical system in Figure 1 (right), where the nature of the approximation is given by the linear map in the center of Figure 1 (i.e., the a coordinate corresponds to $x + y$, and the b coordinate to z). The linear map is a simulation, in the sense that it transforms the state space of the program into the state space of the linear dynamical system so that every step in the loop has a corresponding step in the linear dynamical system.

```
1   z := 1
2   while (x ≥ 0 ∧ y ≥ 0) do
3      w := 3w + x + 1
4      if ((x - y) % 2 == 0):
5         x := x - z
6      else:
7         y := y - z
```

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$$

$$- - - - - - - - - - - - - - \rightarrow \begin{bmatrix} a \\ b \end{bmatrix} := \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$
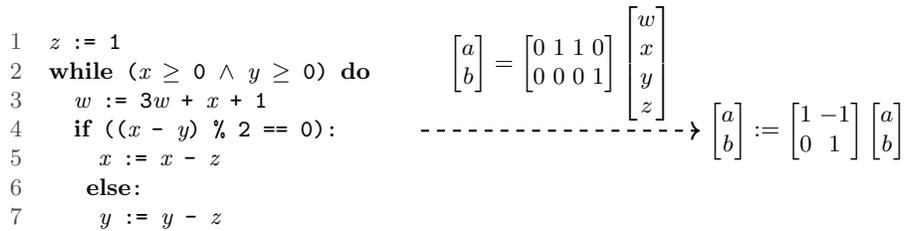
Fig. 1: Over-approximation of a loop by a linear dynamical system.

Next, we compute the image of the guard of the loop $(x \geq 0 \wedge y \geq 0)$ under the simulation, which yields $a \geq 0$ (corresponding to the constraint $x + y \geq 0$ over the original program variables). We can compute a closed form for this constraint holding on the kth iteration of the loop by exponentiating the dynamics matrix of the linear dynamical system, multiplying on the left by the row vector corresponding to the constraint, and on the right by the simulation:

$$\underbrace{\begin{bmatrix} 1\ 0 \end{bmatrix}}_{\text{Constraint}} \underbrace{\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}^k}_{\text{Dynamics}} \underbrace{\begin{bmatrix} 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix}}_{\text{Simulation}} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = (x + y) - kz.$$

We then analyze the asymptotic behavior of the closed form:

$$\text{As } k \to \infty, (x + y) - kz \to \begin{cases} -\infty & \text{if } z > 0 \\ x + y & \text{if } z = 0 \\ \infty & \text{if } z < 0 \end{cases}$$

We conclude that $z > 0 \vee (x + y) < 0$ is a sufficient condition for the loop to terminate.   ⌐

The paper is organized as follows. To serve as the class of "linear models" of loops, we introduce *deterministic affine transition system*s (DATS), a computational model that generalizes affine dynamical systems. Section 3 shows that any loop expressed as a linear integer arithmetic formula has a *DATS-reflection*, which is a best representation of the behavior of the loop as a DATS. Moreover, this holds for a restricted class of DATS with rational eigenvalues. Section 4 shows that for a linear map f with integer eigenvalues and a linear integer arithmetic formula G, there is a linear integer arithmetic formula that holds exactly for those states x such that $G(f^k(x))$ holds for all but finitely many $k \in \mathbb{N}$. Section 5 brings the results together, showing that the analysis of a DATS with rational eigenvalues can be reduced to the analysis of a linear dynamical system with integer eigenvalues. The fact that DATS-reflections are *best* implies monotonicity of the analysis. Finally, in Section 6, we demonstrate experimentally that the analysis can be successfully applied to general programs, using

the framework of algebraic termination analysis [34] to lift our loop analysis to a whole-program conditional termination analysis. Some proofs are omitted for space, but may be found in the extended version of this paper [33].

2   Preliminaries

This paper assumes familiarity with linear algebra – see for example [19]. We recall some basic definitions below.

In the following, a **linear space** refers to a finite-dimensional linear space over the field of rational numbers $\mathbb{Q}$. For V a linear space and $U \subseteq V$, $span(U)$ is the linear space generated by U; i.e., the smallest linear subspace of V that contains U. An **affine subspace** of a linear space V is the image of a linear subspace of V under a translation (i.e., a set of the form $\{v + v_0 : v \in U\}$ for some linear subspace $U \subseteq V$ and some $v_0 \in V$). For any scalar $a \in \mathbb{Q}$, and any linear space V, we use $\underline{a}$ to denote the linear map $\underline{a} : V \to V$ that maps $v \mapsto av$ (in particular, $\underline{1}$ is the identity). A **linear functional** on a linear space V is a linear map $V \to \mathbb{Q}$; the set of all linear functionals on V forms a linear space called the **dual space** of V, denoted $V^\star$. A linear map $f : V_1 \to V_2$ induces a dual linear map $f^\star : V_2^\star \to V_1^\star$ where $f^\star(g) \triangleq g \circ f$. For any linear space V, V is naturally isomorphic to $V^{\star\star}$, where the isomorphism maps $x \mapsto \lambda f : V^\star.f(x)$.

Let V be a linear space. A linear map $f : V \to V$ is associated with a **characteristic polynomial** $p_f(x)$, which is defined to be the determinant of $(xI - A_f)$, where A_f is a matrix representation of f with respect to some basis (the choice of which is irrelevant). Define the **spectrum** (set of eigenvalues) of f to be the set of (possibly complex) roots of its characteristic polynomial, $spec(f) \triangleq \{\lambda \in \mathbb{C} : p_f(\lambda) = 0\}$. We say that f has **rational spectrum** if $spec(f) \subseteq \mathbb{Q}$; equivalently (by the spectral theorem – see e.g. [19, Ch. 6, Theorem 7]):

- There is a basis $\{x_1, ..., x_n\}$ for V consisting of *generalized (right) eigenvectors*, satisfying $(f - \underline{\lambda_i})^{r_i}(x_i) = 0$ for some $\lambda_i \in spec(f)$ and some $r_i \geq 1$ (r_i is called the *rank* of x_i)
- There is a basis $\{g_1, ..., g_n\}$ for $V^\star$ consisting of *generalized left eigenvectors*, satisfying $g_i \circ (f - \underline{\lambda_i})^{r_i} = 0$ for some $\lambda_i \in spec(f)$ and some $r_i \geq 1$

It is possible to determine whether a linear map has rational spectrum (and compute the basis of eigenvectors for V and $V^\star$) in polynomial time by computing its characteristic polynomial [15], factoring it [22], and checking whether each factor is linear.

The syntax of linear integer arithmetic (LIA) is given as follows:

$$x \in \mathsf{Variable}$$
$$n \in \mathbb{Z}$$
$$t \in \mathsf{Term} ::= x \mid n \mid n \cdot t \mid t_1 + t_2$$
$$F \in \mathsf{Formula} ::= t_1 \leq t_2 \mid (n \mid t) \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \neg F \mid \exists x.F \mid \forall x.F$$

Let $X \subseteq \mathsf{Variable}$ be a set of variables. A **valuation** over X is a map $v : X \to \mathbb{Z}$. If F is a formula whose free variables range over X and v is a valuation over X, then we say that v satisfies F (written $v \models F$) if the formula F is true when interpreted over the standard model of the integers, using v to interpret the free variables. We write $F \models G$ if every valuation that satisfies F also satisfies G.

2.1 Transition systems

A **transition system** T is a pair $T = \langle S_T, R_T \rangle$ where S_T is a set of states and $R_T \subseteq S_T \times S_T$ is a transition relation. Within this paper, we shall assume that the state space of any transition system is a finite-dimensional linear space (over $\mathbb{Q}$). We write $x \to_T x'$ to denote that the pair $\langle x, x' \rangle$ belongs to R_T. We define the **domain** of a transition system T, $\mathrm{dom}(T) \triangleq \{x \in S_T : \exists x'. x \to_T x'\}$, to be the set of states that have a T-successor. We define the ω-**domain** $\mathrm{dom}^\omega(T)$ of T to be the set of states from which there exist infinite T-computations:

$$\mathrm{dom}^\omega(T) \triangleq \{x_0 \in S_T : \exists x_1, x_2, ... \text{ such that } x_0 \to_T x_1 \to_T x_2 \to_T \cdots\} \ .$$

A **transition formula** $F(X, X')$ is an LIA formula whose free variables range over a designated finite set of variables X and a set of "primed copies" $X' = \{x' : x \in X\}$. For example, a transition formula that represents the body of the loop in Figure 1 is

$$
\begin{aligned}
x \geq 0 \wedge y \geq 0 \wedge w' = 3w + x + 1 \wedge z' = z \\
\wedge \left(\begin{array}{l} ((2 \mid x - y) \wedge x' = x - z \wedge y' = y) \\ \vee \, (\neg(2 \mid x - y) \wedge y' = y - z \wedge x' = x) \end{array} \right)
\end{aligned}
\tag{1}
$$

We use **TF** to denote the set of transition formulas. A transition formula $F(X, X')$ defines a transition system where the state space is the set of functions $X \to \mathbb{Q}$, and where $v \to_F v'$ if and only if both (1) v and v' map each $x \in X$ to an integer and (2) $[v, v'] \models F$, where $[v, v']$ denotes the valuation that maps each $x \in X$ to $v(x)$ and each $x' \in S'$ to $v'(x)$. Defining the state space of F to be $X \to \mathbb{Q}$ rather than $X \to \mathbb{Z}$ is a technical convenience ($X \to \mathbb{Q} \cong \mathbb{Q}^{|X|}$ is a linear space), but does not materially affect the results of this paper since only (integral) valuations are involved in transitions.

Let $T = \langle S_T, R_T \rangle$ be a transition system. We say that T is:

- **linear** if R_T is a linear subspace of $S_T \times S_T$,
- **affine** if R_T is an affine subspace of $S_T \times S_T$,
- **deterministic** if $x \to_T x'_1$ and $x \to_T x'_2$ implies $x'_1 = x'_2$
- **total** if for all $x \in S_T$ there exists some $x' \in S_T$ with $x \to_T x'$

For example, the transition system T with transition relation

$$
R_T \triangleq \left\{ \left\langle \begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix} \right\rangle : \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}
$$

is deterministic and affine, but not linear or total. The transition system U with transition relation

$$R_U \triangleq \left\{ \left\langle \begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix} \right\rangle : [1\ 1] \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right\}$$

is total, linear (and affine), but not deterministic. The classical notion of a **linear dynamical system**—a transition system where the state evolves according to a linear map—corresponds to a *total, deterministic, linear* transition system. Similarly, an **affine dynamical system** is a transition system that is total, deterministic, and affine.

For any map $s : X \to Y$, and any relation $R \subseteq X \times X$, define the image of R under s to be the relation $s[R] = \{\langle s(x), s(x') \rangle : \langle x, x' \rangle \in R\}$. For any relation $R \subseteq Y \times Y$, define the inverse image of R under s to be the relation $s^{-1}[R] = \{\langle x, x' \rangle : \langle s(x), s(x') \rangle \in R\}$. Let $T = \langle S_T, R_T \rangle$ and $U = \langle S_U, R_U \rangle$ be transition systems. We say that a linear map $s : S_T \to S_U$ is a **linear simulation from T to U**, and write $s : T \to U$, if for all $x \to_T x'$, we have $s(x) \to_U s(x')$. Observe that the following are equivalent: (1) s is a simulation, (2) $s[R_T] \subseteq R_U$, and (3) $R_T \subseteq s^{-1}[R_U]$.

An example of a simulation between a transition formula and a linear dynamical system is given in Figure 1. In fact, there are many linear dynamical systems that over-approximate this loop; however, the simulation and linear dynamical system given in Figure 1 is its *best abstraction*.

To formalize the meaning of *best abstractions*, it is convenient to use the language of category theory [17]. Any class of transition systems defines a category, where the objects are transitions systems of that class, and the arrows are linear simulations between them. We use boldface letters (**L**inear, **A**ffine, **D**eterministic, **T**otal) to denote categories of transition systems (e.g., **DATS** denotes the category of **D**eterministic **A**ffine **T**ransition **S**ystems).

If T is a transition system and **C** is a category of transition systems, a **C-abstraction** of T is a pair $\langle U, s \rangle$ consisting of a transition system U belonging to **C** and a linear simulation $s : T \to U$. A **C-reflection** of T is a **C**-abstraction that satisfies a universal property among **C**-abstractions of T: for any **C**-abstraction $\langle V, t \rangle$ of T there exists a unique simulation $\bar{t} : U \to V$ such that $\bar{t} \circ s = t$; i.e., the following diagram commutes:

$$
\begin{array}{ccc}
 & & V \\
 & \overset{t}{\nearrow} & \uparrow \bar{t} \\
T & \underset{s}{\longrightarrow} & U
\end{array}
$$

If **D** is a category of transition systems and **C** is a subcategory such that every transition system in **D** has a **C**-reflection, we say that **C** is a **reflective subcategory** of **D**.

Our ultimate goal is to bring techniques from linear dynamical systems to bear on transition formulas. Figure 1 gives an example of a program and its linear

dynamical system reflection. Unfortunately, such reflections do not exist for *all* transition formulas, which motivates our investigation of alternative models.

Proposition 1. *The transition formula* $x' = x \wedge x = 0$ *has no* **TDATS**-*reflection.*

Proof. Let F be the 1-dimensional transition formula $x' = x \wedge x = 0$. For a contradiction, suppose that $\langle A, s \rangle$ is a **TDATS**-reflection of F. Since F contains the origin, then so must the transition relation of A, and so A is linear. Next, consider that for any $\lambda \in \mathbb{Q}$, we have the simulation $id : F \to A_\lambda$, where id is the identity function and $A_\lambda = \langle \mathbb{Q}, x \mapsto \lambda x \rangle$. Since $\langle A, s \rangle$ is a reflection of F, for any λ, there is some t_λ such that $t_\lambda : A \to A_\lambda$ and $id = t_\lambda \circ s$. Since t_λ is a simulation, we have $\lambda t_\lambda = A_\lambda \circ t_\lambda = t_\lambda \circ A$. Since $id = t_\lambda \circ s$, we must have t_λ non-zero, and so t_λ is a left eigenvector of A with eigenvalue λ. Since this holds for all λ, A must have infinitely many eigenvalues, a contradiction.

3   Linear abstractions of transition formulas

Proposition 1 shows that not every transition formula has a total deterministic affine reflection. In the following we show that *totality* is the only barrier: every transition formula has a (computable) **DATS**-reflection. Moreover, we show that every transition formula has a *rational spectrum* **DATS** ($\mathbb{Q}$-**DATS**)-reflection, a restricted class of **DATS** that generalizes affine maps $x \mapsto A\mathbf{x} + \mathbf{b}$ where A has rational eigenvalues. The restriction on eigenvalues makes it easier to reason about the termination behavior of $\mathbb{Q}$-**DATS**.

In the remainder of this section, we show that every transition formula has a $\mathbb{Q}$-**DATS**-reflection by establishing a chain of reflective subcategories:

$$\mathbf{TF} \xrightarrow{\text{Lemma 1}} \mathbf{ATS} \xrightarrow{\text{Lemma 3}} \mathbf{DATS} \xrightarrow{\text{Corollary 1}} \mathbb{Q}\text{-}\mathbf{DATS}$$

The fact that $\mathbb{Q}$-**DATS** is a reflective subcategory of **TF** then follows from the fact that a reflective subcategory of a reflective subcategory is reflective.

3.1   Affine abstractions of transition formulas

Let $F(X, X')$ be a transition formula. The **affine hull** of F, denoted $\textit{aff}(F)$, is the smallest affine set $\textit{aff}(F) \subseteq (X \cup X') \to \mathbb{Q} \cong (X \to \mathbb{Q}) \times (X \to \mathbb{Q})$ that contains all of the models of F. Reps et al. give an algorithm that can be used to compute $\textit{aff}(F)$, by using an SMT solver to sample a set of generators [26].

Lemma 1. *Let* $F(X, X')$ *be a transition formula. The affine hull of F (considered as a transition system) is the best affine abstraction of F (where the simulation from F to $\textit{aff}(F)$ is the identity).*

Example 2. Consider the example program in Figure 1. Letting F denote the transition formula corresponding to the program, $aff(F)$ can be represented as the solutions to the constraints

$$\begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix} \begin{bmatrix} w' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 3\ 1\ 0\ \ 0 \\ 0\ 1\ 1\ -1 \\ 0\ 0\ 0\ \ 1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

Notice that $aff(F)$ is 4-dimensional and has a transition relation defined by 3 constraints, and thus is *not* deterministic. The next step is to find a suitable projection onto a lower-dimensional space so that the resulting transition system is deterministic.   ⌋

3.2   Reflections via the dual space

This section presents a key technical tool that will be used in the next two subsections to prove the existence of reflections. For any transition system T, an abstraction $\langle U, s \rangle$ of T consisting of a transition system U and a simulation $s : S_T \to S_U$ induces a subspace of $S_T^\star$, which is the range of the dual map $s^\star$ (i.e., the set of all linear functionals on S_T of the form $g \circ s$ where $g \in S_U^\star$). The essential idea is we can apply this in reverse: any subspace Λ of $S_T^\star$ induces a transition system U and a simulation $s : T \to U$ that satisfies a universal property among all abstractions $\langle V, v \rangle$ of T where the range of $v^\star$ is contained in Λ. We will now formalize this idea.

Let T be a transition system, and let Λ be a subspace of $S_T^\star$. Define $\alpha_\Lambda(T)$ to be the pair $\alpha_\Lambda(T) \triangleq \langle U, s \rangle$ consisting of a transition system U and a linear simulation $s : T \to U$ where

- $s : S_T \to \Lambda^\star$ sends each $x \in S_T$ to $\lambda f : \Lambda . f(x)$
- $S_U \triangleq \Lambda^\star$, and $R_U \triangleq s[R_T] = \{\langle s(x), s(x') \rangle : \langle x, x' \rangle \in R_T \}$

Lemma 2 (Dual space simulation).   *Let T be a transition system, let Λ be a subspace of $S_T^\star$, and let $\langle U, s \rangle = \alpha_\Lambda(T)$. Suppose that Z is a transition system and $z : T \to Z$ is a simulation such that the range of $z^\star$ is contained in Λ. Then there exists a unique simulation $\overline{z} : U \to Z$ such that $\overline{z} \circ s = z$.*

Proof. The high-level intuition is that since the range of $z^\star$ is contained in Λ, we may consider it to be a map $z^\star : S_Z^\star \to \Lambda$; dualizing again, we get a map $z^{\star\star} : \Lambda^\star \to S_Z^{\star\star}$, whose domain is S_U and codomain is (isomorphic to) S_Z.

More formally, let $j : S_Z \to S_Z^{\star\star}$ be the natural isomorphism between S_Z and $S_Z^{\star\star}$ defined by $j(y) \triangleq \lambda g : S_Z^\star . g(y)$. Define $\overline{z} : \Lambda^\star \to S_Z$ by

$$\overline{z}(h) \triangleq j^{-1}(\lambda g : S_Z^\star . h(g \circ z)) \ .$$

First we show that $\overline{z} \circ s = z$. Let $x \in S_Z$. Then we have

$$
\begin{aligned}
(\overline{z} \circ s)(x) &= \overline{z}(s(x)) \\
&= j^{-1}(\lambda g : S_Z^\star.(s(x))(g \circ z)) \\
&= j^{-1}(\lambda g : S_Z^\star.(\lambda f : \Lambda.f(x))(g \circ z)) \\
&= j^{-1}(\lambda g : S_Z^\star.g(z(x))) \\
&= z(x) \ .
\end{aligned}
$$

Next we show that $\overline{z}$ is a simulation. Suppose $y \to_U y'$. Since $R_U = s[R_T]$, there is some $x, x' \in S_T$ such that $x \to_T x'$, $s(x) = y$, and $s(x') = y'$. Since $z : T \to Z$ is a simulation, we have that $z(x) \to_Z z(x)$, and so $\overline{z}(s(x)) \to_Z \overline{z}(s(x'))$, and we may conclude that $\overline{z}(y) \to_Z \overline{z}(y')$.

Finally, observe that s is surjective, and therefore the solution to the equation $\overline{z} \circ s = z$ is unique.

We conclude this section by illustrating how to compute the function α for affine transition systems. Suppose that T is an affine transition system of dimension n. We can represent states in S_T by vectors in $\mathbb{Q}^n$, and the transition relation R_T by a finite set of transitions $B \subseteq \mathbb{Q}^n \times \mathbb{Q}^n$ that generates R_T (i.e., $R_T = \mathit{aff}(B)$). Suppose that Λ is an m-dimensional subspace of $S_T^\star$; elements of $S_T^\star$ can be represented by n-dimensional row vectors, and Λ can be represented by a basis $\mathbf{f}_1^\mathsf{T}, \ldots, \mathbf{f}_m^\mathsf{T}$. We can compute a representation of $\langle U, s \rangle = \alpha_\Lambda(T)$ as follows. The elements of $S_U = \Lambda^\star$ can be represented by m-dimensional vectors (with respect to the basis $g_1, \ldots, g_m$ such that g_i is the linear map that sends $\mathbf{f}_j^\mathsf{T}$ to 1 if $i = j$ and to 0 otherwise). The simulation s can be represented by the $m \times n$ matrix where the ith row is $\mathbf{f}_i^\mathsf{T}$. Finally, the transition relation R_U can be represented by a set of generators $\{\langle s(\mathbf{x}), s(\mathbf{x}') \rangle : \langle \mathbf{x}, \mathbf{x}' \rangle \in B\}$.

3.3   Determinization

In this section, we show that any transition system operating over a finite-dimensional vector space has a best deterministic abstraction, and give an algorithm for computing the best deterministic affine abstraction (or *determinization*) of an affine transition system.

Towards an application of Lemma 2, we seek to characterize the determinization of a transition system by a space of functionals on its state space. For any linear space V and space of functionals Λ on V, define an equivalence relation $\equiv_\Lambda$ on V by $x \equiv_\Lambda y$ iff $f(x) = f(y)$ for all $f \in \Lambda$. If T is a transition system and Λ, Λ' are spaces of functionals on S_T, we say that T is (Λ, Λ')-**deterministic** if for all $x_1, x_2 \ x_1', x_2'$ such that $x_1 \equiv_\Lambda x_2$, $x_1 \to_T x_1'$, and $x_2 \to_T x_2'$, then we also have $x_1' \equiv_{\Lambda'} x_1'$. Observe that if D is a deterministic transition system and $d : T \to D$ is a simulation, then T must be (Λ_d, Λ_d)-deterministic, where Λ_d is the range of the dual map $d^\star$.

For any T and Λ, define $\mathsf{Det}(T, \Lambda) \triangleq \{f : T \text{ is } (\Lambda, \{f\})\text{-deterministic}\}$ to be the greatest set of functionals such that T is $(\Lambda, \mathsf{Det}(T, \Lambda))$-deterministic.

Observe that $\mathsf{Det}(T, -)$ is a monotone operator on the complete lattice of linear subspaces of $S_T^\star$ (i.e., if $\Lambda_1 \subseteq \Lambda_2$ then $\mathsf{Det}(T, \Lambda_1) \subseteq \mathsf{Det}(T, \Lambda_2)$, since Λ_1 induces a coarser equivalence relation than Λ_2). By the Knaster-Tarski fixpoint theorem [28], $\mathsf{Det}(T, -)$ has a greatest fixpoint, which we denote by $\mathsf{Det}(T)$. Then we have that T is $(\mathsf{Det}(T), \mathsf{Det}(T))$-deterministic, and $\mathsf{Det}(T)$ contains every space Λ such that T is (Λ, Λ)-deterministic.

Lemma 3 (Determinization). *For any transition system T, $\alpha_{\mathsf{Det}(T)}(T)$ is a deterministic reflection of T.*

Proof. Let $\langle D, d \rangle \triangleq \alpha_{\mathsf{Det}(T)}(T)$. First, we show that D is deterministic. Suppose that $y \to_D y_1'$ and $y \to_D y_2'$; we must show that $y_1' = y_2'$. Since R_D is defined to be $d[R_T]$, there must be x_1, x_2, x_1', and x_2' in S_T such that $x_1 \to_T x_1'$, $x_2 \to_T x_2'$, $d(x_1) = d(x_2) = y$, $d(x_1') = y_1'$, and $d(x_2') = y_2$. Since $d(x_1) = d(x_2)$, we have $(\lambda f : \mathsf{Det}(T).f(x_1)) = (\lambda f : \mathsf{Det}(T).f(x_2))$, and therefore $x_1 \equiv_{\mathsf{Det}(T)} x_2$. We thus have $x_1' \equiv_{\mathsf{Det}(T, \mathsf{Det}(T))} x_2'$, and since $\mathsf{Det}(T, \mathsf{Det}(T)) = \mathsf{Det}(T)$, we have $y_1' = d(x_1') = d(x_2') = y_2'$.

It remains to show that $\langle D, d \rangle$ is a deterministic *reflection* of T. Suppose that $\langle U, u \rangle$ is another deterministic abstraction of T. Define G to be the range of $u^\star$. Since U is deterministic, we must have $G \subseteq \mathsf{Det}(T, G)$, and since $\mathsf{Det}(T)$ is the greatest fixpoint of $\mathsf{Det}(T, -)$ we have $G \subseteq \mathsf{Det}(T)$. By Lemma 2, there is a unique linear simulation $\bar{u} : D \to U$ such that $\bar{u} \circ d = u$.

If a transition system T is affine, then its determinization can be computed in polynomial time. Fixing a basis for the state space S_T (of some dimension n), we can represent the transition relation of T in the form $R_T = \{\langle \mathbf{x}, \mathbf{x}' \rangle : A\mathbf{x}' = B\mathbf{x} + \mathbf{c}\}$ where $A, B \in \mathbb{Q}^{m \times n}$ and $\mathbf{c} \in \mathbb{Q}^m$ (for some m). We can represent functionals on S_T by n-dimensional vectors, where the vector $\mathbf{v} \in \mathbb{Q}^n$ corresponds to the functional that maps $\mathbf{u} \mapsto \mathbf{v}^\intercal \mathbf{u}$. A linear space of functionals Λ can be represented by a system of linear equations $\Lambda = \{\mathbf{x} : M\mathbf{x} = 0\}$. The ith row $\mathbf{a}_i^\intercal \mathbf{v} = \mathbf{b}_i^\intercal \mathbf{u} + c_i$, of the system of equations $A\mathbf{x}' = B\mathbf{x} + \mathbf{c}$ can be read as "T is $(\{\mathbf{b}_i^\intercal\}, \{\mathbf{a}_i^\intercal\})$-deterministic." Thus, the functionals $\mathbf{f}^\intercal$ such that T is $(\Lambda, \{\mathbf{f}^\intercal\})$-deterministic are those that can be written as a linear combination of the rows of A such that the corresponding linear combination of the rows of B belongs to Λ; i.e.,

$$\mathsf{Det}(\{\langle \mathbf{x}, \mathbf{x}' \rangle : A\mathbf{x}' = B\mathbf{x} + \mathbf{c}\}, \{\mathbf{f} : M\mathbf{f} = 0\}) = \{\mathbf{d} : \exists \mathbf{y}.MB^\intercal \mathbf{y} = 0 \wedge A^\intercal \mathbf{y} = \mathbf{d}\} \ .$$

A representation of $\mathsf{Det}(T, \Lambda)$ can be computed in polynomial time using Gaussian elimination. Since the lattice of linear subspaces of $S_T^\star$ has height n, the greatest fixpoint of $\mathsf{Det}(T, -)$ can be computed in polynomial time.

Example 3. Continuing the example from Figure 1 and Example 2, we consider the determinization of the affine transition system in Eq (2). The rows of the matrix on the left-hand side correspond to generators for $\mathsf{Det}(\mathit{aff}(F), \mathbb{Q}^{4\star})$:

$$\mathsf{Det}(\mathit{aff}(F), \mathbb{Q}^{4\star}) = \mathit{span}(\{[1\ 0\ 0\ 0], [0\ 1\ 1\ 0], [0\ 0\ 0\ 1]\})$$

$$\mathsf{Det}(\mathit{aff}(F), \mathsf{Det}(\mathit{aff}(F), \mathbb{Q}^{4\star})) = \mathit{span}(\{[0\ 1\ 1\ 0], [0\ 0\ 0\ 1]\})$$

which is the greatest fixpoint $\mathsf{Det}(\textit{aff}(F))$. Intuitively: after one step of $\textit{aff}(F)$, the values of w, $x + y$, and z are affine functions of the input; after two steps $x + y$ and z are affine functions of the input but w is not, since the value of w on the second step depends upon the value of x in the first, and x is not an affine function of the input.

This yields the deterministic reflection $\langle D, d \rangle$ (pictured in Figure 1) where

$$R_D = \left\{ \left\langle \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} a' \\ b' \end{bmatrix} \right\rangle : \begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\} \qquad \text{and} \qquad d = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \lrcorner$$

3.4 Rational-spectrum reflections of DATS

In this section, we define rational-spectrum **DATS** and show that every **DATS** has a rational-spectrum-reflection.

In the following, it is convenient to work with transition systems that are linear rather than affine. We will prove that every deterministic *linear* transition system has a best abstraction with rational spectrum. The result extends to the affine case through the use of *homogenization*: i.e., we embed a (non-empty) affine transition system into a linear transition system with one additional dimension, such that if we fix that dimension to be 1 then we recover the affine transition system. If the transition relation of a **DATS** is represented in the form $A\mathbf{x}' = B\mathbf{x} + \mathbf{c}$, then its homogenization is simply

$$\begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ y \end{bmatrix} = \begin{bmatrix} B & \mathbf{c} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} \ .$$

For a **DATS** T, we use $\mathsf{homog}(T)$ to denote the pair $\langle L, h \rangle$, consisting the **DLTS** L resulting from homogenization and the affine simulation $h : T \to L$ that maps each $\mathbf{x} \in S_T$ to $\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$ (i.e., the affine simulation h formalizes the idea that if we fix the extra dimension y to be 1, we recover the original **DATS** T).

Let T be a deterministic linear transition system. Since our goal is to analyze the asymptotic behavior of T, and all long-running behaviors of T reside entirely within $\text{dom}^\omega(T)$, we are interested in the structure of $\text{dom}^\omega(T)$ and T's behavior on this set. First, we observe that $\text{dom}^\omega(T)$ is a linear subspace of S_T and is computable. For any k, let T^k denote the linear transition system whose transition relation is the k-fold composition of the transition relation of R. Consider the descending sequence of linear spaces

$$\text{dom}(T) \supseteq \text{dom}(T^2) \supseteq \text{dom}(T^3) \supseteq \dots$$

(i.e., the set of states from which there are T computations of length 1, length 2, length 3, ...). Since the space S_T is finite dimensional, this sequence must stabilize at some k. Since the states in $\text{dom}(T^k)$ have T-computations of any length and T is deterministic, we have that $\text{dom}(T^k)$ is precisely $\text{dom}^\omega(T)$.

Since T is total on $\text{dom}^\omega(T)$ and the successor of a state in $\text{dom}^\omega(T)$ must also belong to $\text{dom}^\omega(T)$, T defines a linear map $T|_\omega : \text{dom}^\omega(T) \to \text{dom}^\omega(T)$. In

this way, we can essentially reduce asymptotic analysis of **DATS** to asymptotic analysis of linear dynamical systems. The asymptotic analysis of linear dynamical systems developed in Sections 4 and 5 requires rational eigenvalues; thus we are interested in **DATS** T such that $T|_\omega$ has rational eigenvalues. With this in mind, we define $spec(T) = spec(T|_\omega)$, and say that T **has rational spectrum** if $spec(T) \subseteq \mathbb{Q}$. Define $\mathbb{Q}$-**DLTS** to be the subcategory of **DLTS** with rational spectrum, and $\mathbb{Q}$-**DATS** to be the subcategory of **DATS** whose homogenization lies in $\mathbb{Q}$-**DLTS**.

Example 4. Consider the **DLTS** T with

$$R_T \triangleq \left\{ \left\langle \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \right\rangle : \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\}$$

The bottom-most equation corresponds to a constraint that only vectors where the x and y coordinates are equal have successors, so we have:

$$\mathrm{dom}(T) = \left\{ \begin{bmatrix} x & y & z \end{bmatrix}^\intercal : x = y \right\}$$

Supposing that the x and y coordinates are equal in some pre-state, they are equal in the post-state exactly when $z = 0$, so we have

$$\mathrm{dom}(T^2) = \left\{ \begin{bmatrix} x & y & z \end{bmatrix}^\intercal : x = y \wedge z = 0 \right\}$$

It is easy to check that $\mathrm{dom}(T^3) = \mathrm{dom}(T^2)$, and therefore $\mathrm{dom}^\omega(T) = \mathrm{dom}(T^2)$. The vector $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^\intercal$ is a basis for $\mathrm{dom}^\omega(T)$, and the matrix representation of $T|_\omega$ with respect to this basis is $\begin{bmatrix} 2 \end{bmatrix}$ (i.e., $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^\intercal \rightarrow_T \begin{bmatrix} 2 & 2 & 0 \end{bmatrix}^\intercal$). Thus we can see $spec(T) = \{2\}$, and T is a $\mathbb{Q}$-**DLTS**.   ⌟

Towards an application of Lemma 2, define the **generalized rational eigenspace** of a DLTS T to be

$$E_\mathbb{Q}(T) \triangleq span\left(\left\{ f \in S_T^\star : \exists \lambda \in \mathbb{Q}, \exists r \in \mathbb{N}^+ . f \circ (T|_\omega - \underline{\lambda})^r = 0 \right\} \right).$$

Lemma 4. *Let T be a DLTS, and define $\langle Q, q \rangle \triangleq \alpha_{E_\mathbb{Q}(T)}(T)$. Then for any $\mathbb{Q}$-**DLTS** U and any simulation $s : T \rightarrow U$, there is a unique simulation $\overline{s} : Q \rightarrow U$ such that $\overline{s} \circ q = s$.*

While $\alpha_{E_\mathbb{Q}(T)}(T)$ satisfies a universal property for $\mathbb{Q}$-**DLTS**, it does not necessary belong to $\mathbb{Q}$-**DLTS** itself because it need not be deterministic. However, by iterative interleaving of Lemma 4 and determinization as shown in Algorithm 1, we arrive at a $\mathbb{Q}$-**DLTS**-reflection. Example 5 demonstrates how we calculate a $\mathbb{Q}$-**DLTS**-reflection of a particular **DLTS**.

Example 5. Consider the **DLTS** T with transition relation

$$R_T \triangleq \left\{ \left\langle \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}, \begin{bmatrix} w' \\ x' \\ y' \\ z' \end{bmatrix} \right\rangle : \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \right\}$$

We can calculate the ω-domain of T $\mathrm{dom}^\omega(T) = \left\{ \begin{bmatrix} w\ x\ y\ z \end{bmatrix}^\mathsf{T} : w = x \right\}$, which has a basis $B = \begin{bmatrix} 1\ 1\ 0\ 0 \end{bmatrix}^\mathsf{T}, \begin{bmatrix} 0\ 0\ 1\ 0 \end{bmatrix}^\mathsf{T}, \begin{bmatrix} 0\ 0\ 0\ 1 \end{bmatrix}^\mathsf{T}$. With respect to B, $T|_\omega$ corresponds to the matrix

$$T|_\omega = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

and so we have $spec(T) = \{2, i, -i\}$. We may calculate $E_{\mathbb{Q}}(T)$ by finding (generalized) left eigenvectors with eigenvalue 2, the only rational number in $spec(T)$:

$$E_{\mathbb{Q}}(T) = \left\{ \mathbf{v}^\mathsf{T} : \mathbf{v}^\mathsf{T} \underbrace{\begin{bmatrix} 1\,0\,0 \\ 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}}_{B} \left(\underbrace{\begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}}_{T|_\omega} - \underbrace{\begin{bmatrix} 2\,0\,0 \\ 0\,2\,0 \\ 0\,0\,2 \end{bmatrix}}_{2I} \right) = 0 \right\}$$

$$= span(\begin{bmatrix} 1\ 1\ 0\ 0 \end{bmatrix}, \begin{bmatrix} 1\ -1\ 0\ 0 \end{bmatrix})$$

Finally, we have $\langle Q, q \rangle = \alpha_{E_{\mathbb{Q}}(T)}(T)$, where

$$R_Q = \left\{ \left\langle \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} a' \\ b' \end{bmatrix} \right\rangle : \begin{bmatrix} 1\,0 \\ 0\,1 \\ 0\,0 \end{bmatrix} \begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} 2\,0 \\ 0\,0 \\ 0\,1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\} \qquad q = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

Q is deterministic and has rational spectrum, so $\langle Q, q \rangle$ is a $\mathbb{Q}$-**DLTS**-reflection of T. ⌟

Theorem 1. *For any deterministic linear transition system, Algorithm 1 computes a $\mathbb{Q}$-**DLTS**-reflection.*

Finally, by homogenization and Theorem 1, we conclude with the desired result:

Corollary 1. $\mathbb{Q}$-**DATS** *is a reflective subcategory of* **DATS**.

4   Asymptotic Analysis of Linear Dynamical Systems

This section is concerned with analyzing the behavior of loops of the form

$$\textbf{while } (G(\mathbf{x})) \textbf{ do } \mathbf{x} := A\mathbf{x} \ ,$$

Input   : A **DLTS** T.
Output : $\mathbb{Q}$-**DLTS**-reflection of T
1 $U \leftarrow T$;
2 $s \leftarrow \lambda x.x$;   /* *Invariant: s is a simulation from T to U* */
3 **while** $spec(U|_\omega) \nsubseteq \mathbb{Q}$ **do**
4 | $\langle Q, q \rangle \leftarrow \alpha_{E_{\mathbb{Q}}(U)}(U)$;   /* *Lemma 4* */
5 | $\langle U, d \rangle \leftarrow \alpha_{\mathsf{Det}(Q)}(Q)$;   /* *Lemma 3* */
6 | $s \leftarrow d \circ q \circ s$;
7 **return** $\langle U, s \rangle$

Algorithm 1: Computation of a $\mathbb{Q}$-**DLTS**-reflection of a **DLTS**

where the $G(\mathbf{x})$ is an LIA formula and A is a matrix with integer spectrum. Our goal is to capture the asymptotic behavior of iterating the map A on an initial state $\mathbf{x}_0$ with respect to the formula G. Specifically, we show that

Theorem 2. *For any LIA formula G and any matrix A with integer spectrum, there is a periodic sequence of LIA formulas $H_0, H_1, H_2, \ldots$ such that for any initial state $\mathbf{x}_0 \in \mathbb{Q}^n$, there exists K such that for any $k > K$, $G(A^k\mathbf{x}_0)$ holds if and only if $H_k(\mathbf{x}_0)$ does.*

Recall that an infinite sequence $H_0, H_1, H_2, \ldots$ is *periodic* if it is of the form

$$(H_0, H_1, \ldots, H_P)^\omega \triangleq H_0, H_1, \ldots, H_P, H_0, H_1, \ldots, H_P, \ldots$$

We call the periodic sequence $(H_0, H_1, \ldots, H_P)^\omega$ the *characteristic sequence* of the guard formula G with respect to dynamics matrix A, and denote it by $\chi(G, A)$. Note that $G(A^k\mathbf{x}_0)$ holds for all but finitely many k exactly when $\bigwedge_{i=0}^{P} H_i(\mathbf{x}_0)$ holds.

In the remainder of this section, we show how to compute characteristic sequences. Let G be an LIA formula and let A be a matrix with integer spectrum. To begin, we compute a quantifier-formula G' that is equivalent to G (using, for example, Cooper's algorithm [7]). We define $\chi(G', A)$ by recursion on the structure of G'. For the logical connectives $\wedge$, $\vee$, and $\neg$, characteristic sequences are defined pointwise:

$$\chi(\neg H, A) \triangleq (\neg(\chi(H, A)_0), \neg(\chi(H, A)_1), \ldots)$$

$$\chi(H_1 \wedge H_2, A) \triangleq (\chi(H_1, A)_0 \wedge \chi(H_2, A)_0, \chi(H_1, A)_1 \wedge \chi(H_2, A)_1, \ldots)$$

$$\chi(H_1 \vee H_2, A) \triangleq (\chi(H_1, A)_0 \vee \chi(H_2, A)_0, \chi(H_1, A)_1 \vee \chi(H_2, A)_1, \ldots)$$

It remains to show how χ acts on atomic formulas, which take the form of inequalities $t_1 \leq t_2$ and divisibility constraints $n \mid t$. An important fact that we employ in both cases is that for any linear term $\mathbf{c}^\mathsf{T}\mathbf{x}$ over the variables $\mathbf{x}$, we can compute a closed form for $\mathbf{c}^\mathsf{T} A^k(\mathbf{x})$ by symbolically exponentiating A. Since (by assumption) A has integer eigenvalues, this closed form has the form $\frac{1}{Q}(p(\mathbf{x}, k))$ where $Q \in \mathbb{N}$ and p is an **integer exponential-polynomial term**, which takes the form

$$\lambda_1^k k^{d_1} \mathbf{a}_1^\mathsf{T}\mathbf{x} + \cdots + \lambda_m^k k^{d_m} \mathbf{a}_m^\mathsf{T}\mathbf{x} \tag{3}$$

where $\lambda_i \in spec(A)$, $d_i \in \mathbb{N}$, and $\mathbf{a}_i \in \mathbb{Z}^n$.[1]

Characteristic sequences for inequalities Our method for computing characteristic sequences for inequalities is a variation of Tiwari's method for deciding termination of linear loops with real eigenvalues [29].

First, suppose that $\mathbf{p}(\mathbf{x}, k)$ is an integer exponential-polynomial of the form in Eq. (3) such that each λ_i is a *positive* integer. Further suppose that the summands are ordered by asymptotic growth, with the dominant term appearing earliest in the list; i.e., for $i < j$ we have either $\lambda_i > \lambda_j$, or $\lambda_i = \lambda_j$ and $d_i > d_j$. If we imagine that the variables $\mathbf{x}$ are fixed to some $\mathbf{x}_0 \in \mathbb{Z}^n$, then we see that $p(\mathbf{x}_0, k)$ is either identically zero or has finitely many zeros, and therefore its sign is eventually stable. Furthermore, the sign of $p(\mathbf{x}_0, k)$ as k tends to ∞ is simply the sign of its *dominant term* – that is, the sign of $\mathbf{a}_i^\mathsf{T}\mathbf{x}_0$ for the least i such that $\mathbf{a}_i^\mathsf{T}\mathbf{x}_0$ is non-zero. Thus, we may define a function DTA that maps any exponential-polynomial term $p(\mathbf{x}, k)$ (with positive integral λ_i) to an LIA formula such that for any $\mathbf{x}_0 \in \mathbb{Z}^n$, $\mathbf{x}_0 \models \mathsf{DTA}(p)$ holds if and only if $\mathbf{p}(\mathbf{x}_0, k)$ is eventually non-negative ($\mathbf{p}(\mathbf{x}_0, k) \geq 0$ for all but finitely many $k \in \mathbb{N}$). DTA is defined as follows:

$$\mathsf{DTA}(0) \triangleq true$$
$$\mathsf{DTA}(\lambda^k k^d \mathbf{a}^\mathsf{T}\mathbf{x} + p) \triangleq \mathbf{a}^\mathsf{T}\mathbf{x} \geq 1 \vee (\mathbf{a}^\mathsf{T}\mathbf{x} = 0 \wedge \mathsf{DTA}(p))$$

Finally, we define the characteristic sequence of an inequality atom as follows. An inequality $t_1 \leq t_2$ over the variables $\mathbf{x}$ can be written as $\mathbf{c}^\mathsf{T}\mathbf{x} + d \geq 0$ for $\mathbf{c} \in \mathbb{Z}^n$ and $d \in \mathbb{Z}$. Let $\frac{1}{Q_{even}}p_{even}(\mathbf{x}, k)$ and $\frac{1}{Q_{odd}}p_{odd}(\mathbf{x}, k)$ be the closed forms of $\mathbf{c}^\mathsf{T}A^{2k}(\mathbf{x})$ and $\mathbf{c}^\mathsf{T}A^{2k+1}(\mathbf{x})$, respectively; by splitting into "even" and "odd" cases, we ensure that the exponential-polynomial terms $p_{even}(\mathbf{x}, k)$ and $p_{odd}(\mathbf{x}, k)$ have only *positive* λ_i and thus are amenable to the dominant term analysis DTA described above. Then we define:

$$\chi\left(\mathbf{c}^\mathsf{T}\mathbf{x} + d \geq 0, A\right) \triangleq (\mathsf{DTA}(p_{even}(\mathbf{x}, k) + dQ_{even}), \mathsf{DTA}(p_{odd}(\mathbf{x}, k) + dQ_{odd}))^\omega$$

Example 6. Consider the matrix A and its exponential A^k below:

$$A\left(\begin{bmatrix} x' \\ y' \\ z' \\ a' \\ b' \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ a \\ b \end{bmatrix}$$

$$A^k\left(\begin{bmatrix} x \\ y \\ z \\ a \\ b \end{bmatrix}\right) = \begin{bmatrix} 1 & k & \frac{k(k-1)}{2} & 0 & 0 \\ 0 & 1 & k & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & (-3)^k & 0 \\ 0 & 0 & 0 & 0 & 2^k \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(zk^2 + (2y - z)k + 2x) \\ zk + y \\ z \\ (-3)^k a \\ 2^k b \end{bmatrix}$$

---

[1] Technically, we have $\frac{1}{Q}(\lambda_1^k k^{d_1} \mathbf{a}_1^\mathsf{T} + \cdots + \lambda_m^k k^{d_m} \mathbf{a}_m^\mathsf{T}) = \mathbf{c}^\mathsf{T}A^k\mathbf{x}$ for all k greater than rank of the highest-rank generalized eigenvector of 0, but since we are only interested in the asymptotic behavior of A we can disregard the first steps of the computation.

First we compute the characteristic sequence $\chi(x \geq 0, A)$. Applying the dominant term analysis of the closed form of x yields

$$\mathsf{DTA}\left(zk^2 + (2y - z)\,k + x\right) = \begin{pmatrix} z > 0 \\ \vee\ (z = 0 \wedge 2y - z > 0) \\ \vee\ (z = 0 \wedge 2y - z = 0 \wedge x \geq 0) \end{pmatrix},$$

Since the closed form involves only positive exponential terms, we need not split into an even and odd case, and we simply have:

$$\chi(x \geq 0, A) = (z > 0 \vee (z = 0 \wedge 2y - z > 0) \vee (z = 0 \wedge 2y - z = 0 \wedge x \geq 0))^\omega$$

Next we compute the characteristic sequence $\chi(a - b \geq 0, A)$, which does require a case split. Applying dominant term analysis of the closed form of $(a - b)$ yields

$$\mathsf{DTA}(a \cdot (-3)^{2k} - b \cdot 2^{2k}) = a > 0 \vee (a = 0 \wedge -b \geq 0)$$
$$\mathsf{DTA}(a \cdot (-3)^{2k+1} - b \cdot 2^{2k+1}) = -a > 0 \vee (-a = 0 \wedge -b \geq 0) .$$

and thus we have

$$\chi(a - b \geq 0, A) = (a > 0 \vee (a = 0 \wedge -b \geq 0), -a > 0 \vee (-a = 0 \wedge -b \geq 0))^\omega .\quad \lrcorner$$

Characteristic sequences for divisibility atoms Last we show how to define χ for divisibility atoms $n \mid t$. Write the term t as $\mathbf{c}^\mathsf{T}\mathbf{x} + d$ and let the closed form of $\mathbf{c}^\mathsf{T} A^k(\mathbf{x})$ be

$$\frac{1}{Q}(\lambda_1^k k^{d_1} \mathbf{a}_1^\mathsf{T}\mathbf{x} + \cdots + \lambda_m^k k^{d_m} \mathbf{a}_m^\mathsf{T}\mathbf{x}) .$$

The formula $n \mid \mathbf{c}^\mathsf{T} A^k(\mathbf{x}) + d$ is equivalent to $Qn \mid \lambda_1^k k^{d_1} \mathbf{a}_1^\mathsf{T}\mathbf{x} + \cdots + \lambda_m^k k^{d_m} \mathbf{a}_m^\mathsf{T}\mathbf{x} + Qd$. For any i, the sequence $\langle \lambda_i^k k^{d_i} \bmod Qn \rangle_{k=0}^\infty$ is ultimately periodic, since (1) $\langle k \bmod Qn \rangle_{k=0}^\infty = (0, 1, \ldots, Qn - 1)^\omega$, (2) $\langle \lambda_i^k \bmod Qn \rangle_{k=0}^\infty$ is ultimately periodic (with period and transient length bounded above by $Qn)^2$, and (3) ultimately periodic sequences are closed under pointwise product. It follows that for each i, there is a periodic sequence of integers $\langle z_{i,k} \rangle_{k=0}^\infty$ that agrees with $\langle \lambda_i^k k^{d_i} \bmod Qn \rangle_{k=0}^\infty$ on all but finitely many terms. Finally, we take

$$\chi(n \mid t, A) \triangleq \langle Qn \mid z_{1,k}\mathbf{a}_1^\mathsf{T}\mathbf{x} + \cdots + z_{m,k}\mathbf{a}_m^\mathsf{T}\mathbf{x} + Qd \rangle_{k=0}^\infty .$$

Example 7. Consider matrix A and the closed form of its exponents below

$$A\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad A^k\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5^k \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

---

[2] An infinite sequence $s_0, s_1, s_2, \ldots$ is *ultimately periodic*, if there exists N such that $s_N, s_{N+1}, s_{N+2}, \ldots$ is a periodic sequence. We call N the transient length of this sequence.

We show the characteristic sequences for some divisibility atoms w.r.t A:

$$\chi(3 \mid x, A) = (3 \mid x, 3 \mid x + y, 3 \mid x + 2y)^{\omega}$$
$$\chi(3 \mid x + 2, A) = (3 \mid x + 2, 3 \mid x + y + 2, 3 \mid x + 2y + 2)^{\omega}$$
$$\chi(3 \mid z, A) = (3 \mid z, 3 \mid 2z)^{\omega} \qquad \qquad \lrcorner$$

5   A conditional termination analysis for programs

This section demonstrates how the results from Sections 3 and 4 can be combined to yield a conditional termination analysis that applies to general programs.

Integer-spectrum restriction for $\mathbb{Q}$-DLTS Section 3 gives a way to compute a $\mathbb{Q}$-**DATS**-reflection of any transition formula. Yet the analysis we developed in Section 4 only applies to linear dynamical systems with integer spectrum. We now show how to bridge the gap. Let V be a $\mathbb{Q}$-**DATS**. As discussed in Section 3.4, we may homogenize V to obtain a $\mathbb{Q}$-**DLTS** T. Define $\mathbb{Z}(T)$ to be the space spanned by the generalized (right) eigenvectors of $T|_{\omega}$ that correspond to integer eigenvalues:

$$\mathbb{Z}(T) \triangleq span(\{x \in \text{dom}^{\omega}(T) : \exists r \in \mathbb{N}^{+}, \lambda \in \mathbb{Z}.(T|_{\omega} - \underline{\lambda})^{r}(x) = 0\})$$

Since $\mathbb{Z}(T)$ is invariant under $T|_{\omega}$ and thus T, T defines a linear map $T|_{\mathbb{Z}} : \mathbb{Z}(T) \to \mathbb{Z}(T)$, and by construction $T|_{\mathbb{Z}}$ has integer spectrum. The following lemma justifies the restriction of our attention to the subspace $\mathbb{Z}(T)$.

Lemma 5. *Let F be a transition formula, let $\langle V, s \rangle$ be a $\mathbb{Q}$-**DATS**-reflection of F, and let $\langle T, h \rangle = \mathsf{homog}(V)$. For any state $v \in \text{dom}^{\omega}(F)$, we have $h(s(v)) \in \mathbb{Z}(T)$.*

Example 8. The following loop computes the number of trailing 0's in the binary representation of integer x and its corresponding transition formula:

```
1   c := 0
2   while (x % 2 == 0) do
3       x = x / 2
4       c = c + 1
```

$$F(x, c, x', c') = \begin{pmatrix} (2 \mid x) \\ \wedge \ (x - 1 \le 2x' \wedge 2x' \le x) \\ \wedge \ (c' = c + 1) \end{pmatrix}$$

The homogenization of the $\mathbb{Q}$-**DATS**-reflection of F is the $\mathbb{Q}$-**DLTS** T, where:

$$R_{T} \triangleq \left\{ \left\langle \begin{bmatrix} x \\ c \\ h \end{bmatrix}, \begin{bmatrix} x \\ c \\ h \end{bmatrix} \right\rangle : \begin{bmatrix} x' \\ c' \\ h' \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ c \\ h \end{bmatrix} \right\}$$

The ω-domain of T is the whole state space $\mathbb{Q}^{3}$. Since the eigenvector $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\top}$ of the transition matrix corresponds to a non-integer eigenvalue $\frac{1}{2}$, the x-coordinate of states in $\mathbb{Z}(T)$ must be 0; i.e., $\mathbb{Z}(T) = \{(x, c, y) : x = 0\}$. We conclude that $x \neq 0$ is a sufficient condition for the loop to terminate. $\qquad \lrcorner$

Input   : A transition formula $F(\mathbf{x}, \mathbf{x}') \in \mathbf{TF}$ in linear integer arithmetic.
Output : A mortal precondition $mp(F)$ for F.
1 $A \leftarrow \mathit{aff}(F)$;   /* Affine hull [26]; Lemma 1 */
2 $\langle D, d \rangle \leftarrow \alpha_{\mathsf{Det}(A)}(A)$;   /* Determinize; Lemma 3 */
3 $\langle V, q \rangle \leftarrow \mathbb{Q}\text{-}\mathbf{DATS}\text{-reflection of } D$;   /* Algorithm 1 */
4 $v \leftarrow q \circ d$;   /* $\langle V, v \rangle$ is a $\mathbb{Q}$-$\mathbf{DATS}$-reflection of F */
5 $\langle T, h \rangle \leftarrow \mathsf{homog}(V)$;   /* Homogenization of V */
6 $t \leftarrow h \circ v$;   /* t is an affine simulation $F \to T$ */
7 $p \leftarrow$ (any) linear projection of S_T onto $\mathbb{Z}(T)$;
8 $C \leftarrow$ matrix such that $C\mathbf{w} = 0 \iff \mathbf{w} \in \mathbb{Z}(T)$;
9 Let $G(\mathbf{w}) \leftarrow \exists \mathbf{x}, \mathbf{x}'.F(\mathbf{x}, \mathbf{x}') \wedge \mathbf{w} = p(t(\mathbf{x})) \wedge Ct(\mathbf{x}) = 0$;
10 $(H_0(\mathbf{w}), \dots, H_P(\mathbf{w}))^\omega \leftarrow \chi(G(\mathbf{w}), T|_\mathbb{Z})$;   /* Section 4 */
11 **return** $\neg\big(\big(\bigwedge_i H_i(p(t(\mathbf{x})))\big) \wedge Ct(\mathbf{x}) = 0\big)$
Algorithm 2: Procedure for computing $mp(F)$.

The mortal precondition operator Algorithm 2 shows how to compute a mortal precondition for an LIA transition formula $F(\mathbf{x}, \mathbf{x}')$ (i.e., a sufficient condition for which F terminates). The algorithm operates as follows. First, we compute a $\mathbb{Q}$-$\mathbf{DATS}$-reflection of F, and homogenize to get a $\mathbb{Q}$-$\mathbf{DLTS}$ T and an *affine* simulation $t : F \to T$. Let p denote an (arbitrary) projection from S_T onto $\mathbb{Z}(T)$ (so p is a simulation from T to $T|_\mathbb{Z}$). We then compute an LIA formula G which represents the states $\mathbf{w}$ of $T|_\mathbb{Z}$ such that there is some $v \in \mathrm{dom}(F)$ such that $t(v) \in \mathbb{Z}(T)$ and $p(t(v)) = \mathbf{w}$. Letting $(H_0, ..., H_P)^\omega$ be the characteristic sequence $\chi(G, T|_\mathbb{Z})$, we have that for any $v \in \mathrm{dom}^\omega(F)$, $t(v)$ must belong to $\mathbb{Z}(T)$ and $p(t(v))$ satisfies each H_i, so we define

$$mp(F) \triangleq \{v \in S_F : t(v) \notin \mathbb{Z}(T) \text{ or } v \not\models \bigwedge_i H_i(p(t(\mathbf{x}))).$$

Within the context of the algorithm, we suppose that states of F are represented by n-dimensional vectors, states of T are represented as m-dimensional vectors, and state of $T|_\mathbb{Z}$ are represented as q-dimensional vectors. The affine simulation t is represented in the form $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$, where $A \in \mathbb{Z}^{m \times n}$ and $\mathbf{b} \in \mathbb{Z}^m$, the projection p as a $\mathbb{Z}^{q \times m}$ matrix, and the linear map $T|_\mathbb{Z}$ as a $\mathbb{Q}^{q \times q}$ matrix. The fact that p and t have all integer (rather than rational) entries is without loss of generality, since any simulation can be scaled by the least common denominator of its entries.

Theorem 3 (Soundness). *For any transition formula F, for any state s such that $s \in mp(F)$, we have $s \notin \mathrm{dom}^\omega(F)$.*

Proof. Let T, t, p, C, G, and $H_0, \dots, H_P$ be as in Algorithm 2. We prove the contrapositive: we assume $v \in \mathrm{dom}^\omega(F)$ and prove $v \notin mp(F)$, or equivalently $v \models H_i(p(t(\mathbf{x})))$ for each i and $t(v) \in \mathbb{Z}(T)$. We have $t(v) \in \mathbb{Z}(T)$ by Lemma 5, so it remains only to show that $v \models H_i(p(t(\mathbf{x})))$ for each i.

Since $v \in \mathrm{dom}^\omega(F)$, there exists an infinite trajectory of F starting from v: $v \rightarrow_F v_1 \rightarrow_F v_2 \rightarrow_F \dots$. For any j, let $\mathbf{w}_j = T|_\mathbb{Z}^j(p(t(v)))$. Since $p \circ t$

is an (affine) simulation, we have $\mathbf{w}_j = p(t(v_j))$ for all j. It follows that for any j, we have $[v_j, v_{j+1}] \models F(\mathbf{x}, \mathbf{x}') \wedge \mathbf{w}_j = p(t(\mathbf{x}_j)) \wedge Ct(\mathbf{x}_j) = 0$, and so $G(\mathbf{w}_j) = \exists \mathbf{x}, \mathbf{x}'.F(\mathbf{x}, \mathbf{x}') \wedge \mathbf{w}_j = p(t(\mathbf{x})) \wedge Ct(\mathbf{x}) = 0$ holds for all j. By Theorem 2, $H_i(p(t(\mathbf{x})))$ holds for all H_i.

The proof of soundness requires only that we can compute $\mathbb{Q}$-**DATS**-abstractions of transition formulas. The following is the culmination of our development of $\mathbb{Q}$-**DATS**-*reflections*:

Theorem 4 (Monotonicity). *For any transition formulas F_1 and F_2 such that $F_1 \models F_2$, we have $mp(F_2) \models mp(F_1)$.*

The desire for monotonicity is inspired by the principle that *changes to a program should have a predictable impact on its analysis* [34]. Monotonicity guarantees that more information into the analysis always leads to better results—for example, if a user annotates a procedure with pre-conditions or adds loop invariants into the program, our termination analysis can only produce weaker (that is, better) preconditions for termination. Moreover, in the context of this work, monotonicity also guarantees that if we cannot prove termination using the mp operator that we defined, then *any* linear abstraction of the loop has reachable non-terminating states.

6   Evaluation

Section 5 shows how to compute mortal preconditions for transition formulas. Using the framework of algebraic termination analysis [34], we can "lift" the analysis to compute mortal preconditions for whole programs. The essential idea is to compute summaries for loops and procedures in "bottom-up" fashion, apply the mortal precondition operator from Section 5 to each loop body summary, and then propagate the mortal preconditions for the loops back to the entry of the program (see [34] for more details). We can verify that a program terminates by using an SMT solver to check that its mortal precondition is valid.

We have implemented Algorithm 2 as a mortal precondition operator mp_{LR} ("mortal precondition via Linear Reflections") in ComPACT, a tool that implements the termination analysis framework presented in [34]. We compare the performance of our analysis against 2LS [5], Ultimate Automizer [10] and CPAchecker [23], the top three competitors in the termination category of Competition on Software Verification (SV-COMP) 2020.

Experiments are run on a virtual machine with Ubuntu 18.04, with a single-core Intel Core i7-9750H @ 2.60GHz CPU and 8 GB of RAM. All tools were run with a time limit of 10 minutes.

Benchmarks We tested on a suite of 263 programs divided into 4 categories. The `termination` and `recursive` suites contain small programs with challenging termination arguments, while the `polybench` suite contains larger real-world programs that have relatively simple termination arguments. The `termination`

benchmark	#tasks	mp_{LR}		2LS		UAutomizer		CPAChecker	
		#correct	time	#correct	time	#correct	time	#correct	time
termination	171	98	**100.8**	115	1966.0	**161**	4772.2	126	12108.6
recursive	42	4	**51.0**	-	-	**30**	1781.7	23	530.6
polybench	30	**30**	128.3	0	7602.7	0	16241.6	0	4035.8
linear	20	**20**	37.0	6	**17.6**	8	2841.3	3	3470.7
Total	263	152	**317.1**	121	9586.3	**199**	25636.8	152	20145.7

Fig. 2: Termination verification benchmarks; time in seconds.

	#tasks	mp_{LR}		ComPACT-mp_{LR}		ComPACT+mp_{LR}	
		#correct	time	#correct	time	#correct	time
termination	171	98	**100.8**	141	118.4	**146**	114.4
recursive	42	4	**51.0**	31	95.4	**32**	94.6
polybench	30	**30**	128.3	30	179.6	30	179.1
linear	20	**20**	37.0	15	116.5	**20**	65.1
Total	263	152	**317.1**	217	509.9	**228**	453.3

Fig. 3: Comparing mp_{LR} and ComPACT; time in seconds.

category consists of the *non-recursive, terminating* benchmarks from SV-COMP 2020 in the `Termination-MainControlFlow` suite. The `recursive` category consists of the *recursive, terminating* benchmarks from the `recursive` directory and `Termination-MainControlFlow`. Note that 2LS does not handle recursive programs, so we exclude it from the `recursive` category. Finally, we created a new test suite `linear` consisting of programs with terminating linear abstractions. This suite is designed to exercise the capabilities of the mp_{LR}, and includes all examples from Ben-Amram and Genaim's article [1] on multi-phase ranking functions, loops with disjunctive and/or modular arithmetic guards, and loops that model integer division and remainder calculation.

How does our analysis compare with the state-of-the-art? The comparison of ComPACT using the mp_{LR} operator against state-of-the-art termination analysis tools is shown in Figure 2. ComPACT with mp_{LR} is competitive with (but not dominating) leading tools in terms of number of tasks solved across the suite, and uses substantially less time. The mp_{LR} analysis is least successful on the `termination` and `recursive` suites, which are designed to have difficult termination arguments. Most competitive tools use a portfolio of different termination techniques to approach such problems (e.g., Ultimate Automizer synthesizes linear, nested, multi-phase, lexicographic and piecewise ranking functions); we investigate the use of mp_{LR} in a portfolio solver in the following.

ComPACT with mp_{LR} solves all tasks in the `polybench` suite, which contains numerical programs that have simple termination arguments, but which are larger than the SV-COMP tasks. 2LS, UAutomizer, and CPAChecker exhaust

time or memory limits on all tasks. Nested loops are a problematic pattern that appears in these programs, e.g.,

```
for(int i = 0; i < 4096; i += step)
  for (int j = 0; j < 4096; j += step)
    // no modifications to i, j, or step
```

For such loops, mp_{LR} is guaranteed to synthesize a conditional termination argument that is *at least* as weak as `step > 0` (regardless of the contents of the inner loop) by monotonicity and the fact that the loop body formula entails $i < 4096 \wedge i' = i + step \wedge step' = step$. Ultimate Automizer, CPAChecker, and 2LS cannot make such theoretical guarantees.

The `linear` suite demonstrates that mp_{LR} is capable of proving termination of programs that lie outside the boundaries of the other tools.

Can our analysis improve a portfolio solver? We compare mp_{LR} and Com-PACT in Figure 3. The columns correspond to running ComPACT with the following options: excluding the portfolio from [34] (mp_{LR}), including the portfolio but excluding mp_{LR} (ComPACT-mp_{LR}), and including the portfolio and mp_{LR} (ComPACT+mp_{LR}). ComPACT+mp_{LR} can solve 11 additional tasks over ComPACT-mp_{LR} while adding negligible runtime overhead. In fact, adding mp_{LR} to the portfolio *decreases* the amount of time it takes for ComPACT to complete all benchmark suites. Note that the combined tool is successful on the most termination tasks among all the tools we tested, both overall and for each individual suite except the `termination` category.

7   Related work

Termination analysis of linear loops The universal termination problem for linear loops (or *total deterministic affine transition systems*, in the terminology of Section 4) was posed by Tiwari [29]. The case of linear loops over the reals was resolved by Tiwari [29], over the rationals by Braverman [4], and finally over the integers by Hosseini et al. [14]. In principle, we can combine any of these techniques with our algorithm for computing **DATS**-reflections of transition formulas to yield a sound (but incomplete) termination analysis. The significance of computing a **DATS**-reflection (rather than just "some" abstraction) is that is provides an algorithmic completeness result: if it is possible to prove termination of a loop by exhibiting a terminating linear dynamical system that simulates it, the algorithm will prove termination.

The method introduced in Section 4 to compute characteristic sequences of inequalities is based on the method that Tiwari used to prove decidability of the universal termination problem for linear loops with (positive) real spectra [29]. Tiwari's condition of having *real* spectra is strictly more general than the *integer* spectra used by our procedure; requiring that the spectrum be integer allows us express the **DTA** procedure in linear *integer* arithmetic rather than real arithmetic. Similar procedures appear also in [18,12]. We note in particular

that our results in Sections 4 and 5 subsume Frohn and Giesl's decision procedure for universal termination for upper-triangular linear loops [12]; since every rational upper-triangular linear loop has a rational spectrum (and is therefore a $\mathbb{Q}$-**DATS**), the mortal precondition computed for any rational upper-triangular linear loop is valid iff the loop is universally terminating.

Linear abstractions The formulation of "best abstractions" using reflective subcategories is based on the framework developed in [17]. A variation of this method was used in the context of invariant generation, based on computing (weak) reflections of linear rational arithmetic formulas in the category of rational vector addition systems [27]. This paper is the first to apply the idea to termination analysis.

A method for extracting polynomial recurrence (in)equations that are entailed by a transition formula appears in [16]. The algorithm can also be applied to compute a **TDATS**-abstraction of a transition formula. The procedure does not guarantee that the **TDATS**-abstraction is a reflection (*best* abstraction); Proposition 1 demonstrates that no such procedure exists. In this paper, we generalize the model to allow non-total transition systems, and show that best abstractions do exist. The techniques from Section 3 can be used for invariant generation, improving upon the methods of [16].

Kincaid et al. show that the category of linear dynamical systems with *periodic rational* spectrum is a reflective subcategory of the category of linear dynamical systems [18]. A complex number n is periodic rational if n^p is rational for some $p \in \mathbb{Z}^{>0}$. Combining this result with the technique from Section 3 yields the result that the category of **DATS** with periodic rational spectrum is a reflective subcategory of **TF**. The decision procedure from Section 4 extends easily to the periodic rational case, which results in a strictly more powerful decision procedure.

Termination analysis Termination analysis, and in particular conditional termination analysis, has been widely studied. Work on the subject can be divided into practical termination analyses that work on real programs (but offer few theoretical guarantees) [6,8,30,31,32,20,11,13,2], and work on simplified model (such as linear, octagonal, and polyhedral loops) with strong guarantees (but cannot be applied directly to real programs) [25,3,21,1,29,14,4]. This paper aims to help bridge the gap between the two, by showing how to apply analyses for linear loops to general programs, while preserving some of their desirable theoretical properties, in particular monotonicity.

References

1. Ben-Amram, A.M., Genaim, S.: On multiphase-linear ranking functions. In: CAV. pp. 601–620 (2017)
2. Borralleras, C., Brockschmidt, M., Larraz, D., Oliveras, A., Rodríguez-Carbonell, E., Rubio, A.: Proving termination through conditional termination. In: Legay, A., Margaria, T. (eds.) TACAS. pp. 99–117. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)
3. Bradley, A.R., Manna, Z., Sipma, H.B.: Linear ranking with reachability. In: CAV. pp. 491–504 (2005)
4. Braverman, M.: Termination of integer linear programs. In: Ball, T., Jones, R.B. (eds.) CAV. pp. 372–385. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
5. Chen, H., David, C., Kroening, D., Schrammel, P., Wachter, B.: Bit-precise procedure-modular termination analysis. ACM Trans. Program. Lang. Syst. **40**(1), 1:1–1:38 (2018). https://doi.org/10.1145/3121136, `https://doi.org/10.1145/3121136`
6. Cook, B., Gulwani, S., Lev-Ami, T., Rybalchenko, A., Sagiv, M.: Proving conditional termination. In: Gupta, A., Malik, S. (eds.) CAV. pp. 328–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
7. Cooper, D.C.: Theorem proving in arithmetic without multiplication. Machine intelligence **7**(91-99), 300 (1972)
8. Cousot, P., Cousot, R.: An abstract interpretation framework for termination. In: Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. p. 245–258. POPL '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2103656.2103687, `https://doi.org/10.1145/2103656.2103687`
9. Cyphert, J., Breck, J., Kincaid, Z., Reps, T.: Refinement of path expressions for static analysis. Proc. ACM Program. Lang. **3**(POPL) (Jan 2019). https://doi.org/10.1145/3290358, `https://doi.org/10.1145/3290358`
10. Dietsch, D., Heizmann, M., Nutz, A., Schätzle, C., Schüssele, F.: Ultimate taipan with symbolic interpretation and fluid abstractions - (competition contribution). In: Biere, A., Parker, D. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12079, pp. 418–422. Springer (2020). https://doi.org/10.1007/978-3-030-45237-7_32, `https://doi.org/10.1007/978-3-030-45237-7_32`
11. D'Silva, V., Urban, C.: Conflict-driven conditional termination. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV. pp. 271–286. Springer International Publishing, Cham (2015)
12. Frohn, F., Giesl, J.: Termination of triangular integer loops is decidable. In: Dillig, I., Tasiran, S. (eds.) CAV. pp. 426–444. Springer International Publishing, Cham (2019)
13. Ganty, P., Genaim, S.: Proving termination starting from the end. In: CAV. pp. 397–412 (2013)
14. Hosseini, M., Ouaknine, J., Worrell, J.: Termination of linear loops over the integers. In: Baier, C., Chatzigiannakis, I., Flocchini, P., Leonardi, S. (eds.) ICALP. LIPIcs, vol. 132, pp. 118:1–118:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.118, `https://doi.org/10.4230/LIPIcs.ICALP.2019.118`

15. Keller-Gehrig, W.: Fast algorithms for the characteristic polynomial. Theor. Comput. Sci. **36**(2-3), 309–317 (Jun 1985)
16. Kincaid, Z., Cyphert, J., Breck, J., Reps, T.: Non-linear reasoning for invariant synthesis. PACMPL **2(POPL)**, 54:1–54:33 (2018)
17. Kincaid, Z.: Numerical invariants via abstract machines. In: Podelski, A. (ed.) SAS. pp. 24–42. Springer International Publishing, Cham (2018)
18. Kincaid, Z., Breck, J., Cyphert, J., Reps, T.: Closed forms for numerical loops. Proc. ACM Program. Lang. **3**(POPL) (Jan 2019). https://doi.org/10.1145/3290368, `https://doi.org/10.1145/3290368`
19. Lax, P.D.: Linear algebra and its applications. Wiley-Interscience, second edn. (2007)
20. Le, T.C., Qin, S., Chin, W.N.: Termination and non-termination specification inference. In: PLDI. p. 489–498. PLDI '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2737924.2737993, `https://doi.org/10.1145/2737924.2737993`
21. Leike, J., Heizmann, M.: Ranking templates for linear loops. In: TACAS. pp. 172–186 (2014)
22. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (1982)
23. Ott, S.: Implementing a Termination Analysis using Configurable Program Analysis. Master's thesis, University of Passau (2016)
24. Ouaknine, J., Pinto, J.S., Worrell, J.: On termination of integer linear loops. In: SODA. pp. 957–969 (2015)
25. Podelski, A., Rybalchenko, A.: A complete method for the synthesis of linear ranking functions. In: VMCAI. pp. 239–251 (2004)
26. Reps, T., Sagiv, M., Yorsh, G.: Symbolic implementation of the best transformer. In: Steffen, B., Levi, G. (eds.) Verification, Model Checking, and Abstract Interpretation. pp. 252–266. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
27. Silverman, J., Kincaid, Z.: Loop summarization with rational vector addition systems. In: Dillig, I., Tasiran, S. (eds.) CAV. pp. 97–115. Springer International Publishing, Cham (2019)
28. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific journal of Mathematics **5**(2), 285–309 (1955)
29. Tiwari, A.: Termination of linear programs. In: Alur, R., Peled, D.A. (eds.) CAV. pp. 70–82. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
30. Urban, C.: The abstract domain of segmented ranking functions. In: Logozzo, F., Fähndrich, M. (eds.) SAS. pp. 43–62 (2013)
31. Urban, C., Miné, A.: An abstract domain to infer ordinal-valued ranking functions. In: Shao, Z. (ed.) European Symp. on Programming. pp. 412–431. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
32. Urban, C., Miné, A.: A decision tree abstract domain for proving conditional termination. In: Müller-Olm, M., Seidl, H. (eds.) SAS. pp. 302–318. Springer International Publishing, Cham (2014)
33. Zhu, S., Kincaid, Z.: Reflections on termination of linear loops (2021), `https://arxiv.org/abs/2105.13941`
34. Zhu, S., Kincaid, Z.: Termination analysis without the tears (2021)