

Geometric Series as Nontermination Arguments for Linear Lasso Programs

Jan Leike¹ and Matthias Heizmann²

- 1 The Australian National University
Canberra, Australia
jan.leike@anu.edu.au
- 2 University of Freiburg
Freiburg, Germany
heizmann@informatik.uni-freiburg.de

Abstract

We present a new kind of nontermination argument for linear lasso programs, called *geometric nontermination argument*. A geometric nontermination argument is a finite representation of an infinite execution of the form $(\vec{x} + \sum_{i=0}^t \lambda^i \vec{y})_{t \geq 0}$. The existence of this nontermination argument can be stated as a set of nonlinear algebraic constraints. We show that every linear loop program that has a bounded infinite execution also has a geometric nontermination argument. Furthermore, we discuss nonterminating programs that do not have a geometric nontermination argument.

1998 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases Nontermination analysis, Infinite execution, Constraint-based synthesis, Linear lasso program

1 Introduction

The problem of automatically proving termination of programs has been extensively studied. For restricted classes of programs there are methods proving termination [1, 7] and hence nontermination follows from the absence of a termination proof. For broader classes of programs no complete method for proving termination is known or termination is undecidable. Methods that address these broader classes of programs only check the existence of a certain kind of termination argument, e.g. a specific kind of ranking function. The existence of this termination argument proves termination, however the absence of such a termination argument does not imply nontermination and hence these termination analyses cannot be used to prove nontermination.

Analyses for nontermination proceed in a similar manner. They do not check the existence of a general nontermination proof, instead they check for the existence of a certain kind of nontermination argument, e.g. a recurrence set [3, 5] or an underapproximation of the program that does not terminate for any input [2].

In this paper we present a new kind of nontermination argument for linear lasso programs, called *geometric nontermination argument*. A geometric nontermination argument is a finite representation of an infinite execution that can be denoted as a geometric series. The existence of a geometric nontermination argument can be encoded by a set of nonlinear constraints. Over the reals these constraints are decidable. The advantage of our nontermination arguments lies in their simplicity. In contrast to recurrence sets [3, 5], the constraints that state the existence of our geometric nontermination arguments do not contain quanti-

fier alternation and contain only a small number of nonlinear terms. Unlike [2] we do not need a safety checker to compute nontermination arguments.

2 Preliminaries

We consider the following class of programs whose states are real-valued vectors.

► **Definition 1** (Linear lasso program). A (conjunctive) linear lasso program $P = (\text{STEM}, \text{LOOP})$ consists of two binary relations STEM and LOOP , that are each defined by a formula whose free variables are \vec{x} and \vec{x}' and that have the form $A \begin{pmatrix} \vec{x} \\ \vec{x}' \end{pmatrix} \leq \vec{b}$ for some matrix $A \in \mathbb{R}^{n \times m}$ and some vector $\vec{b} \in \mathbb{R}^m$. We call a linear lasso program *linear loop program* if the formula that defines the relation STEM is equivalent to *true*.

► **Definition 2** (Infinite execution). An infinite sequence of states $(\vec{x}_t)_{t \geq 0}$ is an *infinite execution* of the linear lasso program $P = (\text{STEM}, \text{LOOP})$ iff $(\vec{x}_0, \vec{x}_1) \in \text{STEM}$ and $(\vec{x}_t, \vec{x}_{t+1}) \in \text{LOOP}$ for all $t \geq 1$.

3 Geometric Nontermination Arguments

► **Definition 3.** Let $P = (\text{STEM}, \text{LOOP})$ be a linear lasso program such that LOOP is defined by the formula $A \begin{pmatrix} \vec{x} \\ \vec{x}' \end{pmatrix} \leq \vec{b}$. The tuple $N = (\vec{x}_0, \vec{x}_1, \vec{y}, \lambda)$ is called a *geometric nontermination argument* for P iff the following properties hold.

$$\begin{aligned} (\text{domain}) \quad & \vec{x}_0, \vec{x}_1, \vec{y} \in \mathbb{R}^n, \lambda \in \mathbb{R} \text{ and } \lambda > 0. \\ (\text{init}) \quad & (\vec{x}_0, \vec{x}_1) \in \text{STEM} \\ (\text{point}) \quad & A \begin{pmatrix} \vec{x}_1 \\ \vec{x}_1 + \vec{y} \end{pmatrix} \leq \vec{b} \\ (\text{ray}) \quad & A \begin{pmatrix} \vec{y} \\ \lambda \vec{y} \end{pmatrix} \leq \vec{0} \end{aligned}$$

The constraints (init), (point), and (ray) given in Definition 3 are (quantifier free) nonlinear algebraic constraints, the existence of a solution is decidable [6], and hence the existence of a geometric nontermination argument is decidable. We can check the existence of a geometric nontermination argument by passing the constraints of Definition 3 to an SMT solver for nonlinear real arithmetic [4]. If a satisfying assignment is found, this constitutes a nontermination proof in form of an infinite execution according to the following theorem.

► **Theorem 4** (Soundness). *If the conjunctive linear lasso program $P = (\text{STEM}, \text{LOOP})$ has a geometric nontermination argument $N = (\vec{x}_0, \vec{x}_1, \vec{y}, \lambda)$ then P has the following infinite execution.*

$$\vec{x}_0, \vec{x}_1, \vec{x}_1 + \vec{y}, \vec{x}_1 + (1 + \lambda)\vec{y}, \vec{x}_1 + (1 + \lambda + \lambda^2)\vec{y}, \dots$$

Proof. Define $\vec{z}_0 := \vec{x}_0$ and $\vec{z}_t := \vec{x}_1 + \sum_{i=0}^t \lambda^i \vec{y}$. Then $(\vec{z}_t)_{t \geq 0}$ is an infinite execution of P : by (init), $(\vec{z}_0, \vec{z}_1) = (\vec{x}_0, \vec{x}_1) \in \text{STEM}$ and

$$A \begin{pmatrix} \vec{z}_t \\ \vec{z}_{t+1} \end{pmatrix} = A \begin{pmatrix} \vec{x}_1 + \sum_{i=0}^t \lambda^i \vec{y} \\ \vec{x}_1 + \sum_{i=0}^{t+1} \lambda^i \vec{y} \end{pmatrix} = A \begin{pmatrix} \vec{x}_1 \\ \vec{x}_1 + \vec{y} \end{pmatrix} + \sum_{i=0}^t \lambda^i A \begin{pmatrix} \vec{y} \\ \lambda \vec{y} \end{pmatrix} \leq \vec{b} + \sum_{i=0}^t \lambda^i \vec{0} = \vec{b},$$

by (point) and (ray). ◀

► **Example 5.** Consider the linear loop program $P = (\text{true}, \text{LOOP})$ depicted as pseudocode on the left and whose relation $\text{LOOP}(a, b, a', b')$ is defined by the formula depicted on the right.

$$\begin{array}{l} \mathbf{while} \ (a \geq 7): \\ \quad a := b; \\ \quad b := a + 1; \end{array} \quad \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \\ a' \\ b' \end{pmatrix} \leq \begin{pmatrix} 7 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Note that in this example, the relation LOOP is defined by an affine-linear transformation and a guard $a \geq 7$. In general, linear lasso programs are defined with linear constraints, which also allow nondeterministic updates of variables.

For $x_0 = (\frac{7}{8})$, $x_1 = (\frac{7}{8})$, $y = (\frac{1}{1})$ and $\lambda = 1$, the tuple $N = (x_0, x_1, y, \lambda)$ is a geometric nontermination argument and the following sequence of states is an infinite execution of P .

$$\left(\frac{7}{8}\right), \left(\frac{7}{8}\right), \left(\frac{8}{9}\right), \left(\frac{9}{10}\right), \left(\frac{10}{11}\right), \dots$$

We are able to decide the existence of a geometric nontermination argument, however we are not able to decide the existence of an infinite execution because there are programs that have an infinite execution but no geometric nontermination argument as the following example illustrates.

► **Example 6.** The following linear lasso program has an infinite execution, e.g. $\left(\frac{2^t}{3^t}\right)_{t \geq 0}$, but it does not have a geometric nontermination argument.

$$\begin{array}{l} \mathbf{while} \ (a \geq 1 \wedge b \geq 1): \\ \quad a := 2 \cdot a; \\ \quad b := 3 \cdot b; \end{array}$$

4 Bounded Infinite Executions

In this section we show that we can always prove nontermination of linear loop programs if there is a bounded infinite execution.

Let $|\cdot| : \mathbb{R}^n \rightarrow \mathbb{R}$ denote some norm. We call an infinite execution $(\vec{x}_t)_{t \geq 0}$ *bounded* iff there is a real number $d \in \mathbb{R}$ such that for each state its norm is bounded by d , i.e. $|\vec{x}_t| \leq d$ for all t .

► **Lemma 7 (Fixed Point).** *Let $P = (\text{true}, \text{LOOP})$ be a linear loop program. The loop P has a bounded infinite execution if and only if there is a fixed point $\vec{x}^* \in \mathbb{R}^n$ such that $(\vec{x}^*, \vec{x}^*) \in \text{LOOP}$.*

Proof. If there is a fixed point \vec{x}^* , then the loop has the infinite bounded execution $\vec{x}^*, \vec{x}^*, \dots$. Conversely, let $(\vec{x}_t)_{t \geq 0}$ be an infinite bounded execution. Boundedness implies that there is an $d \in \mathbb{R}$ such that $|\vec{x}_t| \leq d$ for all t . Consider the sequence $\vec{z}_k := \frac{1}{k} \sum_{t=1}^k \vec{x}_t$.

$$\begin{aligned} |\vec{z}_k - \vec{z}_{k+1}| &= \left| \frac{1}{k} \sum_{t=1}^k \vec{x}_t - \frac{1}{k+1} \sum_{t=1}^{k+1} \vec{x}_t \right| = \frac{1}{k(k+1)} \left| (k+1) \sum_{t=1}^k \vec{x}_t - k \sum_{t=1}^{k+1} \vec{x}_t \right| \\ &= \frac{1}{k(k+1)} \left| \sum_{t=1}^k \vec{x}_t - k \vec{x}_{k+1} \right| \leq \frac{1}{k(k+1)} \left(\sum_{t=1}^k |\vec{x}_t| + k |\vec{x}_{k+1}| \right) \\ &\leq \frac{1}{k(k+1)} (k \cdot d + k \cdot d) = \frac{2d}{k+1} \rightarrow 0 \text{ as } k \rightarrow \infty. \end{aligned}$$

Hence the sequence $(\vec{z}_k)_{k \geq 1}$ is a Cauchy sequence and thus converges to some $\vec{z}^* \in \mathbb{R}^n$. We will show that \vec{z}^* is the desired fixed point.

For all t , the polyhedron $Q := \{(\vec{x}) \mid A(\vec{x}) \leq b\}$ contains (\vec{x}_t) and is convex. Therefore for all $k \geq 1$,

$$\frac{1}{k} \sum_{t=1}^k (\vec{x}_t) \in Q.$$

Together with

$$\left(\frac{k+1}{k} \vec{z}_{k+1} \right) = \frac{1}{k} \left(\vec{0} \right) + \frac{1}{k} \sum_{t=1}^k (\vec{x}_t)$$

we infer

$$\left(\left(\frac{k+1}{k} \vec{z}_{k+1} \right) - \frac{1}{k} \left(\vec{0} \right) \right) \in Q,$$

and since Q is closed we have

$$\left(\frac{\vec{z}^*}{\vec{z}^*} \right) = \lim_{k \rightarrow \infty} \left(\left(\frac{k+1}{k} \vec{z}_{k+1} \right) - \frac{1}{k} \left(\vec{0} \right) \right) \in Q. \quad \blacktriangleleft$$

Because fixed points give rise to trivial geometric nontermination arguments, we can derive a criterion for the existence of geometric nontermination arguments from Lemma 7.

► **Corollary 8.** *If the linear loop program $P = (\text{true}, \text{LOOP})$ has a bounded infinite execution, then it has a geometric nontermination argument.*

Proof. By Lemma 7 there is a fixed point \vec{x}^* such that $(\vec{x}^*, \vec{x}^*) \in \text{LOOP}$. We choose $\vec{x}_1 = \vec{x}^*$, $\vec{y} = \vec{0}$, and $\lambda = 1$, which satisfies (point) and (ray) and thus is a geometric nontermination argument for P . ◀

► **Example 9.** Note that according to our definition of a linear lasso program, the relation LOOP is a topologically closed set. If we allowed the formula defining LOOP to also contain strict equalities, Lemma 7 no longer holds: the following program is nonterminating and has a bounded infinite execution, but it does not have a fixed point. However, the topological closure of the relation LOOP contains the fixed point $x^* = 0$.

```
while (x > 0):
  x := 1/2 * x;
```

5 Discussion

5.1 Recurrence Sets

Nontermination arguments related to ours are *recurrence sets* [3, 5]. A recurrence set S is a set of states such that

- at least one state of S is in the range of STEM , i.e.

$$\exists \vec{x}, \vec{x}'. (\vec{x}, \vec{x}') \in \text{LOOP} \wedge \vec{x}' \in S, \text{ and}$$

- for each state in S there is at least one LOOP -successor that is in S , i.e.,

$$\forall \vec{x}. \vec{x} \in S \rightarrow \exists \vec{x}' (\vec{x}, \vec{x}') \in \text{LOOP}.$$

If we restrict the form of S to a convex polyhedron, we can encode its existence using algebraic constraints [3, 5] and hence decide the existence of such a recurrence set. However these algebraic constraints are not easy to solve; they contain nonlinear arithmetic and quantifier

alternation that cannot be eliminated with Farkas lemma if the program is nondeterministic. In contrast to these constraints, our constraints (init), (point), and (ray) contain at most one nonlinear term for each dimension of the state space.

However, recurrence sets are more general nontermination arguments than geometric nontermination arguments as shown by the following lemma.

► **Lemma 10.** *Let $P = (\text{STEM}, \text{LOOP})$ be a linear lasso program and $N = (\vec{x}_0, \vec{x}_1, \vec{y}, \lambda)$ be a geometric nontermination argument for P . The following set S is a recurrence set for P .*

$$S = \left\{ \vec{x}_1 + \sum_{i=0}^t \lambda^i \vec{y} \mid t \in \mathbb{N} \right\}$$

Proof. The state \vec{x}_1 is in the range of STEM by (init). Furthermore, for $\vec{x}_1 + \sum_{i=0}^t \lambda^i \vec{y} \in S$, $\vec{x}_1 + \sum_{i=0}^{t+1} \lambda^i \vec{y} \in S$ and $(\vec{x}_1 + \sum_{i=0}^t \lambda^i \vec{y}, \vec{x}_1 + \sum_{i=0}^{t+1} \lambda^i \vec{y}) \in \text{LOOP}$ according to the proof of Theorem 4. ◀

Furthermore, for every geometric nontermination argument $N = (\vec{x}_0, \vec{x}_1, \vec{y}, \lambda)$ there exists a recurrence set S that is a polyhedron.

$$S = \{ \vec{x} \in \mathbb{R}^n \mid \vec{y}^T (\vec{x} - \vec{x}_1) \geq 0 \wedge \bigwedge_{i \in I} \vec{z}_i^T (\vec{x} - \vec{x}_1) = 0 \},$$

where $(\vec{z}_i)_{i \in I}$ is a span of the vector space orthogonal to \vec{y} . (For $\lambda < 1$ we need to add the additional constraint $\vec{y}^T (\vec{x} - \vec{x}_1) \leq \vec{y}^T (\vec{x}_1 + \frac{1}{1-\lambda} \vec{y})$.)

5.2 Integers vs. Reals

A nonterminating program over the reals may terminate over the integers. If we restrict the states of the linear lasso program to integer-valued vectors, then Theorem 4 only holds if we restrict the values for the variables $\vec{x}_0, \vec{x}_1, \vec{y}$, and λ in the constraints (init), (point), and (ray) to integers. Satisfiability of nonlinear arithmetic over the integers is undecidable and we do not know if our constraints fall into a decidable subclass of this problem. However, we may fix the value of λ in advance to a finite set of values. If we do so, we do not have completeness (we may not find every geometric nontermination argument) but we obtain linear arithmetic constraints, which can be solved efficiently.

References

- 1 Mark Braverman. Termination of integer linear programs. In *CAV*, pages 372–385. Springer, 2006.
- 2 Hong-Yi Chen, Byron Cook, Carsten Fuhs, Kaustubh Nimkar, and Peter O’Hearn. Proving nontermination via safety. In *TACAS*, 2014.
- 3 Ashutosh Gupta, Thomas A. Henzinger, Rupak Majumdar, Andrey Rybalchenko, and Ru-Gang Xu. Proving non-termination. In *POPL*, pages 147–158, 2008.
- 4 Dejan Jovanović and Leonardo De Moura. Solving non-linear arithmetic. In *IJCAR*, pages 339–354. Springer, 2012.
- 5 Andrey Rybalchenko. Constraint solving for program verification theory and practice by example. In *CAV*, pages 57–71. Springer, 2010.
- 6 A. Tarski. A decision method for elementary algebra and geometry. Technical report, RAND Corporation, 1951.
- 7 Ashish Tiwari. Termination of linear programs. In *CAV*, pages 70–82. Springer, 2004.