Geometric Series as Nontermination Arguments for Linear Lasso Programs

Jan Leike
The Australian National University

Matthias Heizmann
University of Freiburg
Nontermination Analysis

nonterminating  ==  nonterminating for some input
==  at least one infinite execution

Kinds of Termination Arguments
▶ ranking function
▶ transition invariant
▶ size-change graphs
▶ dependency pair
...
Nontermination Analysis

nonterminating == nonterminating for some input
== at least one infinite execution

Kinds of Termination Arguments

- ranking function
- transition invariant
- size-change graphs
- dependency pair
- ...
Nontermination Analysis

nonterminating \(\implies\) nonterminating for some input
\(\implies\) at least one infinite execution

Kinds of Termination Arguments

- ranking function
- transition invariant
- size-change graphs
- dependency pair
- ...

Kinds of Nontermination Arguments

- recurrence set
- underapproximation which is nonterminating for each input
- ...
- geometric nontermination argument
Geometric Nontermination Argument

witness for existence of infinite execution (of the following form)

\[ x_0, \ x_1, \ x_1 + y, \ x_1 + (1 + \lambda) \cdot y, \ x_1 + (1 + \lambda + \lambda^2) \cdot y, \ldots \]
Geometric Nontermination Argument

witness for existence of infinite execution (of the following form)

\[
x_0, \quad x_1, \quad x_1 + y, \quad x_1 + (1 + \lambda) \cdot y, \quad x_1 + (1 + \lambda + \lambda^2) \cdot y, \quad \ldots
\]
Geometric Nontermination Argument

witness for existence of infinite execution (of the following form)

\[ x_0, \quad x_1, \quad x_1 + y, \quad x_1 + (1 + \lambda \cdot y), \quad x_1 + (1 + \lambda + \lambda^2 \cdot y), \quad \ldots \]

useful in practice

- Benchmark set from

  Brockschmidt, Cook, Fuhs

  Better termination proving through cooperation (CAV 2013)

  contains 181 programs whose nontermination is known, our tool can prove nontermination for 170 of them

- Benchmarks set from Termination Competition 2014
A lasso program $P$ consists of two binary relations $\text{STEM}(x, x')$ and $\text{LOOP}(x, x')$ over a set of states. A sequence of states $s_0, s_1, s_2, s_3, s_4 \ldots$ is called an infinite execution if

- $(s_0, s_1) \in \text{STEM}$, and
- $(s_t, s_{t+1}) \in \text{LOOP}$ for all $t \geq 1$. 

```plaintext
Example:
b := b - 1
while (a ≥ 0) {
    a := a - b
}
```
A lasso program $P$ consists of two binary relations $\text{STEM}(x, x')$ and $\text{LOOP}(x, x')$ over a set of states.

A sequence of states $s_0, s_1, s_2, s_3, s_4 \ldots$ is called an infinite execution if

- $(s_0, s_1) \in \text{STEM}$, and
- $(s_t, s_{t+1}) \in \text{LOOP}$ for all $t \geq 1$.

Example

\[
\begin{array}{c}
b := b - 1 \\
\text{while } (a \geq 0) \{ \\
\quad a := a - b \\
\text{\}}
\end{array}
\]

$\text{STEM}((\frac{a}{b}), (\frac{a'}{b'}))$

$\quad b' = b - 1 \land a' = a$

$\text{LOOP}((\frac{a}{b}), (\frac{a'}{b'}))$

$\quad a \geq 0 \land a' = a - b \land b' = b$

Infinite execution

\[
(\frac{42}{1}), (\frac{42}{0}), (\frac{42}{0}), (\frac{42}{0}), (\frac{42}{0}), \ldots
\]
Preliminary Considerations

a simple case

The lasso program \( P = (\text{STEM}, \text{LOOP}) \) has an execution of the form

\[
\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_1, \mathbf{s}_1, \mathbf{s}_1 \ldots
\]

iff the following formula is satisfiable.

\[
\text{STEM}(\mathbf{s}_0, \mathbf{s}_1) \land \text{LOOP}(\mathbf{s}_1, \mathbf{s}_1)
\]
Preliminary Considerations

a simple case

The lasso program $P = (\text{STEM}, \text{LOOP})$ has an execution of the form

$$s_0, s_1, s_1, s_1, s_1 \ldots$$

iff the following formula is satisfiable.

$$\text{STEM}(s_0, s_1) \land \text{LOOP}(s_1, s_1)$$

Example

```
    b := b - 1
    while (a ≥ 0) {
        a := a - b
    }
```

STEM($((\frac{a}{b}), (\frac{a'}{b'}))$

$b' = b - 1 \land a' = a$

LOOP($((\frac{a}{b}), (\frac{a'}{b'}))$

$a ≥ 0 \land a' = a - b \land b' = b$

$a_0 \mapsto 42 \quad a_1 \mapsto 42$

$b_0 \mapsto 1 \quad b_1 \mapsto 0$

is satisfying assignment
A “difficult” program

```
while (a ≥ 2) {
    a := 2*a + 1
}
```


\[a_0 = 2, \ a_1 = 2, \ a_2 = 5, \ a_3 = 11, \ a_4 = 23, \ a_5 = 47, \ a_6 = 95, \ a_7 = 191, \ldots\]
A “difficult” program

```plaintext
while (a ≥ 2) {
    a := 2*a + 1
}
```

\[ a_0 = 2, \; a_1 = 2, \; a_2 = 5, \; a_3 = 11, \; a_4 = 23, \; a_5 = 47, \; a_6 = 95, \; a_7 = 191, \; \ldots \]

Consider only lasso programs whose relations \texttt{STEM} and \texttt{LOOP} are given by a conjunction of linear inequalities over the reals.
A “difficult” program

while (a ≥ 2) {
    a := 2*a + 1
}

relation LOOP(a, a')

\[
\begin{pmatrix}
-1 & 0 \\
-2 & 1
\end{pmatrix}
\begin{pmatrix}
a \\
a'
\end{pmatrix} \leq 
\begin{pmatrix}
-2 \\
1 \\
-1
\end{pmatrix}
\]

\[a_0 = 2, \ a_1 = 2, \ a_2 = 5, \ a_3 = 11, \ a_4 = 23, \ a_5 = 47, \ a_6 = 95, \ a_7 = 191, \ldots\]

Consider only lasso programs whose relations STEM and LOOP are given by a conjunction of linear inequalities over the reals. We use vectors and matrices to denote conjunctions of linear inequalities. \(A(x') \leq b\)
Geometric Nontermination Argument

Let \( P = (\text{STEM}, \text{LOOP}) \) be a linear lasso program such that \( \text{LOOP} \) is defined by the formula \( A\left(\frac{x}{x'}\right) \leq b \). The tuple \( N = (x_0, x_1, y, \lambda) \) is called a geometric nontermination argument for \( P \) iff the following properties hold.

\[
\begin{align*}
\text{(domain)} & \quad x_0, x_1, y \in \mathbb{R}^n, \lambda \in \mathbb{R} \text{ and } \lambda > 0. \\
\text{(init)} & \quad (x_0, x_1) \in \text{STEM} \\
\text{(point)} & \quad A\left(\frac{x_1}{x_1+y}\right) \leq b \\
\text{(ray)} & \quad A\left(\frac{y}{\lambda y}\right) \leq 0
\end{align*}
\]
Geometric Nontermination Argument

Let \( P = (\text{STEM}, \text{LOOP}) \) be a linear lasso program such that \( \text{LOOP} \) is defined by the formula \( A \left( \begin{array}{c} x \\ x' \end{array} \right) \leq b \). The tuple \( N = (x_0, x_1, y, \lambda) \) is called a geometric nontermination argument for \( P \) iff the following properties hold.

- (domain) \( x_0, x_1, y \in \mathbb{R}^n, \lambda \in \mathbb{R} \) and \( \lambda > 0 \).
- (init) \( (x_0, x_1) \in \text{STEM} \)
- (point) \( A \left( \begin{array}{c} x_1 \\ y \end{array} \right) \leq b \)
- (ray) \( A \left( \begin{array}{c} y \\ \lambda y \end{array} \right) \leq 0 \)

Theorem (Soundness)

If the conjunctive linear lasso program \( P = (\text{STEM}, \text{LOOP}) \) has a geometric nontermination argument \( N = (x_0, x_1, y, \lambda) \) then \( P \) has the following infinite execution.

\[ x_0, \ x_1, \ x_1 + y, \ x_1 + (1 + \lambda) \cdot y, \ x_1 + (1 + \lambda + \lambda^2) \cdot y, \ldots \]
Let $P = (\text{STEM}, \text{LOOP})$ be a linear lasso program such that $\text{LOOP}$ is defined by the formula $A(x_i) \leq b$. The tuple $N = (x_0, x_1, y, \lambda)$ is called a geometric nontermination argument for $P$ iff the following properties hold.

- **(domain)** $x_0, x_1, y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\lambda > 0$.
- **(init)** $(x_0, x_1) \in \text{STEM}$
- **(point)** $A\left(\frac{x_1}{x_1+y}\right) \leq b$
- **(ray)** $A\left(\frac{y}{\lambda y}\right) \leq 0$

We obtain $N = (x_0, x_1, y, \lambda)$ via constraint solving.
Geometric Nontermination Argument

Let $P = (\text{STEM}, \text{LOOP})$ be a linear lasso program such that $\text{LOOP}$ is defined by the formula $A(\frac{x'}{x}) \leq b$. The tuple $N = (x_0, x_1, y, \lambda)$ is called a geometric nontermination argument for $P$ iff the following properties hold.

(domain) $x_0, x_1, y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\lambda > 0$.

(init) $(x_0, x_1) \in \text{STEM}$

(point) $A(\frac{x_1}{x_1+y}) \leq b$

(ray) $A(\frac{y}{\lambda y}) \leq 0$

$x_0, x_1, x_1 + y, x_1 + (1 + \lambda) \cdot y, x_1 + (1 + \lambda + \lambda^2) \cdot y, \ldots$
while (a ≥ 2) {
    a := 2*a + 1
}

relation LOOP(a, a')

\[
\begin{pmatrix}
-1 & 0 \\
-2 & 1 \\
2 & -1
\end{pmatrix}
\begin{pmatrix}
a \\
a'
\end{pmatrix}
\leq
\begin{pmatrix}
-2 \\
1 \\
-1
\end{pmatrix}
\]

**Constraints for Geometric Nontermination Argument**

- **(domain)** $x_0, x_1, y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\lambda > 0$.
- **(init)** $(x_0, x_1) \in \text{STEM}$
- **(point)** $A(x_{i+1} + y) \leq b$
- **(ray)** $A(\lambda y) \leq 0$

For $a_0 = 2$, $a_1 = 2$, $y = 3$ and $\lambda = 2$, the tuple $N = (a_0, a_1, y, \lambda)$ is a geometric nontermination argument and the following sequence of states is an infinite execution of $P$.

$a_0 = 2, a_1 = 2, a_2 = 5, a_3 = 11, a_4 = 23, a_5 = 47, a_6 = 95, a_7 = 191, \ldots$
Theorem (Soundness)

If the conjunctive linear lasso program \( P = (\text{STEM}, \text{LOOP}) \) has a geometric nontermination argument \( N = (x_0, x_1, y, \lambda) \) then \( P \) has the following infinite execution.

\[
x_0, x_1, x_1 + y, x_1 + (1 + \lambda)y, x_1 + (1 + \lambda + \lambda^2)y, \ldots
\]

Proof.

Define \( z_0 := x_0 \) and \( z_t := x_1 + \sum_{i=0}^{t} \lambda^i y \). Then \( (z_t)_{t \geq 0} \) is an infinite execution of \( P \): by (init), \( (z_0, z_1) = (x_0, x_1) \in \text{STEM} \) and

\[
A(z_t^{z_{t+1}}) = A\left(\frac{x_1 + \sum_{i=0}^{t} \lambda^i y}{x_1 + \sum_{i=0}^{t} \lambda^i y}\right) = A\left(\frac{x_1 + y}{x_1 + y}\right) + \sum_{i=0}^{t} \lambda^i A\left(\frac{y}{y}\right) \leq b + \sum_{i=0}^{t} \lambda^i 0 = b,
\]

by (point) and (ray). \( \square \)
infinite execution

\[ x_0, \quad x_1, \quad x_1 + y, \quad x_1 + (1 + \lambda) \cdot y, \quad x_1 + \left( 1 + \lambda + \lambda^2 \right) \cdot y, \quad \ldots \]

closed formula

for \( i \geq 2 \) \[ x_i = x_1 + \frac{\lambda^{i+1} - 1}{\lambda - 1} \cdot y \]
The following linear lasso program has an infinite execution, e.g. $\left(\frac{2^i}{3^i}\right)_{i \geq 0}$, but it does not have a geometric nontermination argument.

```plaintext
while (a ≥ 1 && b ≥ 1 ) {
    a := 2*a
    b := 3*b
}
```
Let \( | \cdot | : \mathbb{R}^n \to \mathbb{R} \) denote some norm. We call an infinite execution \((x_t)_{t \geq 0}\) **bounded** iff there is a real number \(d \in \mathbb{R}\) such that for each state its norm in bounded by \(d\), i.e. \(|x_t| \leq d\) for all \(t\).

**Lemma (Fixed Point)**

Let \(P = (STEM, LOOP)\) be a linear loop program such that \(STEM = id\). The loop \(P\) has a bounded infinite execution if and only if there is a fixed point \(x^* \in \mathbb{R}^n\) such that \((x^*, x^*) \in LOOP\).

**Corollary**

If the linear loop program \(P = (id, LOOP)\) has a bounded infinite execution, then it has a geometric nontermination argument.
Recurrence Set

A recurrence set $S$ is a set of states such that

- at least one state of $S$ is in the range of $\text{STEM}$, i.e.
  \[ \exists x, x'. (x, x') \in \text{STEM} \land x' \in S, \text{ and} \]

- for each state in $S$ there is at least one $\text{LOOP}$-successor that is in $S$, i.e.,
  \[ \forall x. x \in S \rightarrow \exists x'. (x, x') \in \text{LOOP} \land x' \in S. \]

If we restrict the form of $S$ to a convex polyhedron, (i.e.
$S = \bigwedge_i a_i \cdot x \geq d_i$)
we can encode its existence using algebraic constraints.

<table>
<thead>
<tr>
<th>Gupta, Henzinger, Majumdar, Rybalchenko, Xu</th>
<th>Proving non-termination (POPL 2008)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rybalchenko</td>
<td>Constraint solving for program verification theory and practice by example (CAV 2010)</td>
</tr>
</tbody>
</table>
Recurrence Set

Lemma

Let $P = (\text{STEM, LOOP})$ be a linear lasso program and $N = (x_0, x_1, y, \lambda)$ be a geometric nontermination argument for $P$. The following set $S$ is a recurrence set for $P$.

$$S = \left\{ x_1 + \sum_{i=0}^{t} \lambda^i y \mid t \in \mathbb{N} \right\}$$
Integers vs. Reals

Terminating over the Reals $\Rightarrow$ Terminating over the Integers

Constraints for Geometric Nontermination Argument

(domain) $x_0, x_1, y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\lambda > 0$.

(init) $(x_0, x_1) \in \text{STEM}$

(point) $A \left( \frac{x_1}{x_1+y} \right) \leq b$

(ray) $A \left( \frac{y}{\lambda \cdot y} \right) \leq 0$
Future Work

- If \texttt{LOOP} is linear update and \texttt{STEM} is identity then termination is decideable.

<table>
<thead>
<tr>
<th>Ashish Tiwari</th>
<th>Termination of linear programs (CAV 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark Braverman</td>
<td>Termination of integer linear programs (CAV 2006)</td>
</tr>
</tbody>
</table>

Approach: analyze eigenvalues

- Our approach: relations \texttt{LOOP} and \texttt{STEM} given by linear constraints

Can we combine both approaches?
Our tool: LassoRanker

http://ultimate.informatik.uni-freiburg.de/LassoRanker/