On Deciding Constant Runtime of Linear Loops

Florian Frohn, Jürgen Giesl, Peter Giesl, and Nils Lommen

Motivation

Goal: Develop complete decision procedures for programs

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \le x + y \le 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?
- Does the program have constant runtime?

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?
- Does the program have constant runtime?

Applications in . . .

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?
- Does the program have constant runtime?

Applications in ...

- automatic complexity analysis

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?
- Does the program have constant runtime?

Applications in ...

- automatic complexity analysis
- no "approximation" of loops needed in abstract interpretation, symbolic execution, recurrence analysis, loop summarization, ...

Motivation

Goal: Develop complete decision procedures for programs

$$\text{while } (0 \leq x + y \leq 10) \text{ do}$$
$$(x, y) \leftarrow (x + 1, 2 \cdot y)$$

- Does this program terminate?
- How often do we execute the loop?
- Does the program have constant runtime?

Applications in . . .

- automatic complexity analysis
- no "approximation" of loops needed in abstract interpretation, symbolic execution, recurrence analysis, loop summarization, . . .
- synthesis of multiphase-ranking functions (Ben-Amram et al. (SAS 2019))

Analyzing Single-Path Loops

Termination of Linear Programs

Ashish Tiwari*

SRI International
333 Ravenswood Ave, Menlo Park, CA, USA
tiwari@csl.sri.com

Abstract. We show that termination of a class of linear loop programs is decidable. Linear loop programs are discrete-time linear systems with a loop condition governing termination, that is, a while loop with linear assignments. We relate the termination of such a simple loop, on all initial values, to the eigenvectors corresponding to only the positive real eigenvalues of the matrix defining the loop assignments. This characterization of termination is reminiscent of the famous stability theorems in control theory that characterize stability in terms of eigenvalues.

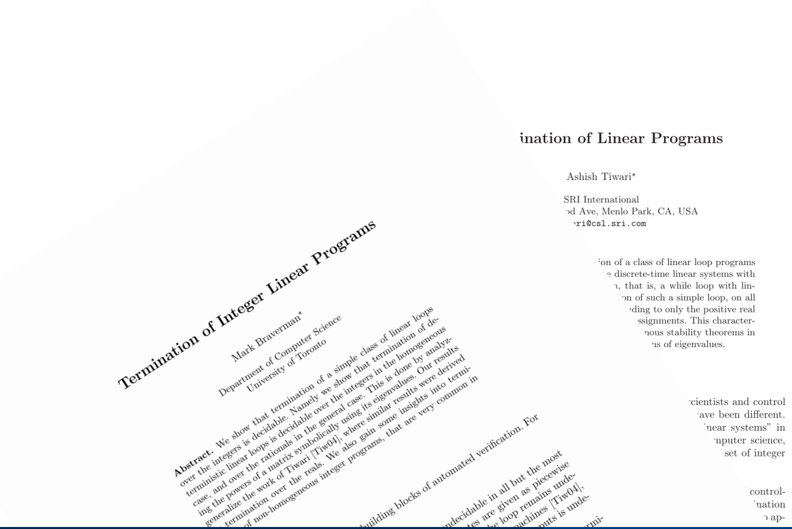
1 Introduction

Dynamical systems have been studied by both computer scientists and control theorists, but both the models and the properties studied have been different. However there is one class of models, called "discrete-time linear systems" in the control world, where there is a considerable overlap. In computer science, these are unconditional `while` loops with linear assignments to a set of integer or rational variables; for example,

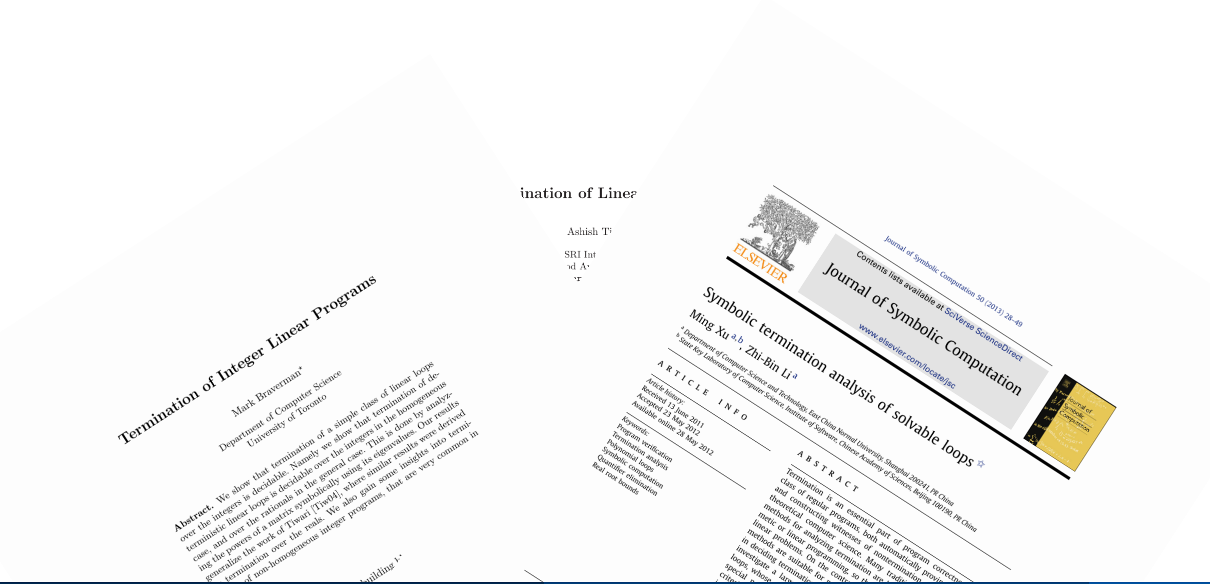
`while` $(true)$ $\{\ x := x - y; y := y\ \}$.

The two communities are interested in different questions: stability and controllability issues in control theory against reachability, invariants, and termination issues in computer science. In recent years, computer scientists have begun to ap-

Analyzing Single-Path Loops



Analyzing Single-Path Loops


Termination of Integer Linear Programs

Mark Braverman*

Department of Computer Science
University of Toronto

Abstract. We show that termination of a simple class of linear loops over the integers is decidable. Namely we show that termination of deterministic linear loops is decidable over the integers in the homogeneous case, and over the rationals in the general case. This is done by analyzing the powers of a matrix symbolically using its eigenvalues. Our results generalize the work of Tiwari [Tiw04], where similar results were derived for termination over the reals. We also gain some insights into termination of non-homogeneous integer programs, that are very common in

ination of Linea

Ashish T

SRI Int

Contents lists available at SciVerse ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Journal of Symbolic Computation 50 (2013) 28–49

ELSEVIER

Symbolic termination analysis of solvable loops ☆

Ming Xu a,b, Zhi-Bin Li a

a Department of Computer Science and Technology, East China Normal University, Shanghai 200241, PR China
b State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PR China

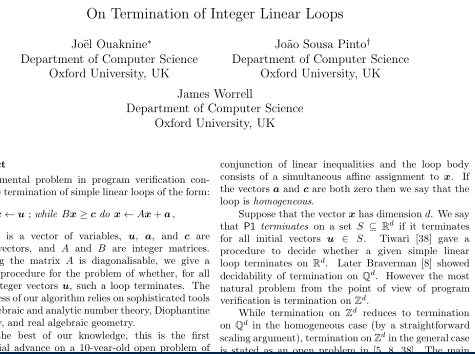
ARTICLE INFO

Article history:
Received 13 June 2011
Accepted 23 May 2012
Available online 28 May 2012

Keywords:
Program verification
Termination analysis
Polynomial loops
Symbolic computation
Quantifier elimination
Real root bounds

ABSTRACT

Analyzing Single-Path Loops



On Termination of Integer Linear Loops

Joël Ouaknine*
Department of Computer Science
Oxford University, UK

João Sousa Pinto†
Department of Computer Science
Oxford University, UK

James Worrell
Department of Computer Science
Oxford University, UK

t
mental problem in program verification con-
termination of simple linear loops of the form:

$\leftarrow \boldsymbol{u}$; *while* $B\boldsymbol{x} \geq \boldsymbol{c}$ *do* $\boldsymbol{x} \leftarrow A\boldsymbol{x} + \boldsymbol{a}$,

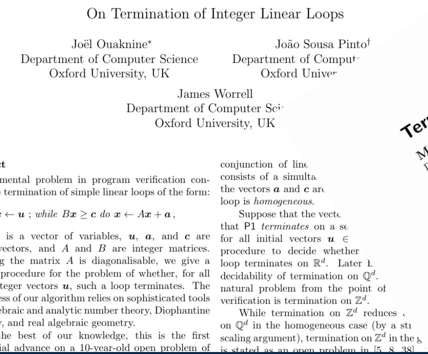
is a vector of variables, $\boldsymbol{u}$, $\boldsymbol{a}$, and $\boldsymbol{c}$ are vectors, and A and B are integer matrices. g the matrix A is diagonalisable, we give a procedure for the problem of whether, for all teger vectors $\boldsymbol{u}$, such a loop terminates. The ess of our algorithm relies on sophisticated tools braic and analytic number theory, Diophantine , and real algebraic geometry.

he best of our knowledge, this is the first ial advance on a 10-year-old open problem of

conjunction of linear inequalities and the loop body consists of a simultaneous affine assignment to $\boldsymbol{x}$. If the vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are both zero then we say that the loop is *homogeneous*.

Suppose that the vector $\boldsymbol{x}$ has dimension d. We say that P1 *terminates* on a set $S \subseteq \mathbb{R}^d$ if it terminates for all initial vectors $\boldsymbol{u} \in S$. Tiwari [38] gave a procedure to decide whether a given simple linear loop terminates on $\mathbb{R}^d$. Later Braverman [8] showed decidability of termination on $\mathbb{Q}^d$. However the most natural problem from the point of view of program verification is termination on $\mathbb{Z}^d$.

While termination on $\mathbb{Z}^d$ reduces to termination on $\mathbb{Q}^d$ in the homogeneous case (by a straightforward scaling argument), termination on $\mathbb{Z}^d$ in the general case is stated as an open problem in [5, 8, 38]. The main



On Termination of Integer Linear Loops

Joël Ouaknine* — Department of Computer Science, Oxford University, UK
João Sousa Pinto† — Department of Comput… Oxford Univer…
James Worrell — Department of Computer Sci… Oxford University, UK

…t

…mental problem in program verification con… termination of simple linear loops of the form:

$\ldots \leftarrow \boldsymbol{u}$; *while* $B\boldsymbol{x} \geq \boldsymbol{c}$ *do* $\boldsymbol{x} \leftarrow A\boldsymbol{x} + \boldsymbol{a}$,

… is a vector of variables, $\boldsymbol{u}$, $\boldsymbol{a}$, and $\boldsymbol{c}$ are …vectors, and A and B are integer matrices. …g the matrix A is diagonalisable, we give a …procedure for the problem of whether, for all …teger vectors $\boldsymbol{u}$, such a loop terminates. The …ess of our algorithm relies on sophisticated tools …ebraic and analytic number theory, Diophantine …, and real algebraic geometry. …he best of our knowledge, this is the first …ial advance on a 10-year-old open problem of

conjunction of line… consists of a simulta… the vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are… loop is *homogeneous*.

Suppose that the vect… that P1 *terminates* on a se… for all initial vectors $\boldsymbol{u} \in$ … procedure to decide whether … loop terminates on $\mathbb{R}^d$. Later … decidability of termination on $\mathbb{Q}^d$. … natural problem from the point of … verification is termination on $\mathbb{Z}^d$.

While termination on $\mathbb{Z}^d$ reduces … on $\mathbb{Q}^d$ in the homogeneous case (by a str… scaling argument), termination on $\mathbb{Z}^d$ in the g… is stated as an open problem in [5, 8, 38]

Termination of Linear Loops over the Integers

Mehran Hosseini
Department of Computer Science, University of Oxford, UK
mehran.hosseini@cs.ox.ac.uk

Joël Ouaknine
Max Planck Institute for Software Systems, Germany
Department of Computer Science, University of Oxford, UK
joel@mpi-sws.org

James Worrell
Department of Computer Science, University of Oxford, UK
jbw@cs.ox.ac.uk

Abstract
We consider the problem of deciding termination of single-path while loops with integer variables, affine updates, and affine guard conditions. The question is whether such a loop terminates on all integer initial values. This problem is known to be decidable for the subclass of loops whose update matrices are diagonalisable, but the general case has remained open since being conjectured decidable by Tiwari in 2004. In this paper we show decidability of determining termination for arbitrary update matrices, confirming Tiwari's conjecture. For the class of loops considered in this … the question of deciding termination on a specific initial value is a longstanding open problem … theory. The key to our decision procedure is in showing how to circumvent the difficulties … deciding termination on a fixed initial value.

… **Classification** Computing methodologies → Algebraic algorithms; Theory of … verification

… Verification, Loop Termination, Linear Integer Programs, Affine

…LP.2019.118

… Theory of Programming

… by ERC grant AVS-ISS (648701). … ISS (648701) and by DFG grant … Science). …208197/1.

On Termination of Integer Linear Loops

Joël Ouaknine* — Department of Computer Science, Oxford University, UK
João Sousa Pinto† — Department of Comput… Oxford Univer…
James Worrell — Department of Computer Sci… Oxford University, UK

…mental problem in program verification con… termination of simple linear loops of the form:

$\ldots \leftarrow \boldsymbol{u}$; *while* $B\boldsymbol{x} \geq \boldsymbol{c}$ *do* $\boldsymbol{x} \leftarrow A\boldsymbol{x} + \boldsymbol{a}$,

… is a vector of variables, $\boldsymbol{u}$, $\boldsymbol{a}$, and $\boldsymbol{c}$ are …vectors, and A and B are integer matrices. …g the matrix A is diagonalisable, we give a …procedure for the problem of whether, for all …teger vectors $\boldsymbol{u}$, such a loop terminates. The …ess of our algorithm relies on sophisticated tools …ebraic and analytic number theory, Diophantine …, and real algebraic geometry. …he best of our knowledge, this is the first …ial advance on a 10-year-old open problem of …

conjunction of line… consists of a simulta… the vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are… loop is *homogeneous*.

Suppose that the vect… that P1 *terminates* on a se… for all initial vectors $\boldsymbol{u} \in$ … procedure to decide whether … loop terminates on $\mathbb{R}^d$. Later … decidability of termination on $\mathbb{Q}^d$. … natural problem from the point of … verification is termination on $\mathbb{Z}^d$.

While termination on $\mathbb{Z}^d$ reduces … on $\mathbb{Q}^d$ in the homogeneous case (by a … scaling argument), termination on $\mathbb{Z}^d$ i… is stated as an open problem in [5…

Termination of Linear Loops …

Mehran Hosseini — Department of Computer Science, Un… mehran.hosseini@cs.ox.ac.uk
Joël Ouaknine — Max Planck In…

Termination of Triangular Integer Loops is Decidable

Florian Frohn[1] and Jürgen Giesl[2]
[1] Max Planck Institute for Informatics, Saarbrücken, Germany — florian.frohn@mpi-inf.mpg.de
[2] LuFG Informatik 2, RWTH Aachen University, Aachen, Germany — giesl@informatik.rwth-aachen.de

Abstract. We consider the problem whether termination of affine integer loops is decidable. Since Tiwari conjectured decidability in 2004 [15], only special cases have been solved [3,4,14]. We complement this work by proving decidability for the case that the update matrix is triangular.

1 Introduction

We consider affine integer loops of the form

while φ **do** $\overline{x} \leftarrow A\,\overline{x} + \overline{a}$.

Here, $A \in \mathbb{Z}^{d \times d}$ for some dimension $d \geq 1$, $\overline{x}$ is a column … different variables $x_1, \ldots, x_d$, $\overline{a} \in \mathbb{Z}^d$, and φ is a con… the form $\alpha > 0$ where $\alpha \in \mathrm{Af}[\overline{x}]$ is an affine express… over $\overline{x}$ (i.e., $\mathrm{Af}[\overline{x}] = \{\overline{c}^T\,\overline{x} + c \mid \overline{c} \in \mathbb{Q}^d, c \in \mathbb{Q}\}$) … where $\overline{0}$ is the vector containing k zeros, … Definition 1 formalizes the intuitive n…

Definition 1 (Termination). …

Analyzing Single-Path Loops

Termination of Polynomial Loops

Florian Frohn[1], Marcel Hark[2(✉)], and Jürgen Giesl[2]

[1] Max Planck Institute for Informatics and Saarland Informatics Campus, Saarbrücken, Germany
[2] LuFG Informatik 2, RWTH Aachen University, Aachen, Germany
marcel.hark@cs.rwth-aachen.de

Abstract. We consider the termination problem for triangular weakly non-linear loops (*twn*-loops) over some ring $\mathcal{S}$ like $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$. Essentially, the guard of such a loop is an arbitrary Boolean formula over (possibly non-linear) polynomial inequations, and the body is a single assignment $\begin{bmatrix} x_1 \\ \cdots \\ x_d \end{bmatrix} \leftarrow \begin{bmatrix} c_1 \cdot x_1 + p_1 \\ \cdots \\ c_d \cdot x_d + p_d \end{bmatrix}$ where each x_i is a variable, $c_i \in \mathcal{S}$, and each p_i is a (possibly non-linear) polynomial over $\mathcal{S}$ and the variables $x_{i+1}, \ldots, x_d$.

We present a reduction from the question of termination to the existential fragment of the first-order theory of $\mathcal{S}$ and $\mathbb{R}$. For loops over $\mathbb{R}$, our reduction entails decidability of termination. For loops over $\mathbb{Z}$ and $\mathbb{Q}$, it proves semi-decidability of non-termination.

Furthermore, we present a transformation to convert certain non-*twn*-loops into *twn*-form. Then the original loop terminates iff the transformed loop terminates over a specific subset of $\mathbb{R}$, which can also be checked via our reduction. This transformation also allows us to prove *tight* complexity bounds for the termination problem for two important classes of loops which can *always* be transformed into *twn*-loops.

1 Introduction

Let $\mathbb{R}_{\mathbb{A}}$ denote the real algebraic numbers. We consider loops of the form

$$\textbf{while } \varphi \textbf{ do } \vec{x} \leftarrow \vec{a}. \quad (1)$$

Termination of Triangular Integer Loops is Decidable

Florian Frohn[1] and Jürgen Giesl[2(✉)]

[1] Max Planck Institute for Informatics, Saarbrücken, Germany
florian.frohn@mpi-inf.mpg.de
[2] LuFG Informatik 2, RWTH Aachen University, Aachen, Germany
giesl@informatik.rwth-aachen.de

Abstract. We consider the problem whether termination of affine integer loops is decidable. Since Tiwari conjectured decidability in 2004 [15], only special cases have been solved [3, 4, 14]. We complement this work by proving decidability for the case that the update matrix is triangular.

1 Introduction

We consider affine integer loops of the form

$$\textbf{while } \varphi \textbf{ do } \overline{x} \leftarrow A\,\overline{x} + \overline{a}.$$

Analyzing Single-Path Loops

- Termination and complexity analysis of programs are undecidable
- **Goal:** Find classes of programs where these problems are decidable
  - Tiwari (CAV 2004)
    - termination is decidable over $\mathbb{R}$
  - Braverman (CAV 2006)
    - termination is decidable over $\mathbb{Q}$
    - termination is decidable over $\mathbb{Z}$ if $\mathbf{b} = \mathbf{0}$
  - Frohn, Giesl, and Hark (CAV 2019; SAS 2020)
    - termination is decidable if *A has real spectrum*
  - Hosseini, Ouaknine, and Worrell (ICALP 2019)
    - termination is decidable over $\mathbb{Z}$

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Analyzing Single-Path Loops

- Termination and complexity analysis of programs are undecidable
- **Goal:** Find classes of programs where these problems are decidable
  - Tiwari (CAV 2004)
    - termination is decidable over $\mathbb{R}$
  - Braverman (CAV 2006)
    - termination is decidable over $\mathbb{Q}$
    - termination is decidable over $\mathbb{Z}$ if $\mathbf{b} = \mathbf{0}$
  - Frohn, Giesl, and Hark (CAV 2019; SAS 2020)
    - termination is decidable if A has real spectrum
  - Hosseini, Ouaknine, and Worrell (ICALP 2019)
    - termination is decidable over $\mathbb{Z}$

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

while $(0 \leq x + y \leq 10)$ do
$(x, y) \leftarrow (x + 1, 2 \cdot y)$

Analyzing Single-Path Loops

- Termination and complexity analysis of programs are undecidable
- **Goal:** Find classes of programs where these problems are decidable
  - Tiwari (CAV 2004)
    - termination is decidable over $\mathbb{R}$
  - Braverman (CAV 2006)
    - termination is decidable over $\mathbb{Q}$
    - termination is decidable over $\mathbb{Z}$ if $\mathbf{b} = \mathbf{0}$
  - Frohn, Giesl, and Hark (CAV 2019; SAS 2020)
    - termination is decidable if A has real spectrum
  - Hosseini, Ouaknine, and Worrell (ICALP 2019)
    - termination is decidable over $\mathbb{Z}$

while $(\varphi(\mathbf{x}))$ do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

while $(0 \le x + y \le 10)$ do
$(x, y) \leftarrow (x + 1, 2 \cdot y)$

while $(0 \le x + y \le 10)$ do
$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Analyzing Single-Path Loops

- Termination and complexity analysis of programs are undecidable
- **Goal:** Find classes of programs where these problems are decidable
  - Tiwari (CAV 2004)
    - termination is decidable over $\mathbb{R}$
  - Braverman (CAV 2006)
    - termination is decidable over $\mathbb{Q}$
    - termination is decidable over $\mathbb{Z}$ if $\mathbf{b} = \mathbf{0}$
  - Frohn, Giesl, and Hark (CAV 2019; SAS 2020)
    - termination is decidable if A has real spectrum
  - Hosseini, Ouaknine, and Worrell (ICALP 2019)
    - termination is decidable over $\mathbb{Z}$

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

while $(0 \leq x + y \leq 10)$ do
$(x, y) \leftarrow (x + 1, 2 \cdot y)$

while $(0 \leq x + y \leq 10)$ do
$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

- **Main Result:** Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum

Constant Runtime of Single-Path Loops (Preprocessing)

- Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum

Constant Runtime of Single-Path Loops (Preprocessing)

- Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum
- Reduce problem to non-negative spectrum by chaining

Constant Runtime of Single-Path Loops (Preprocessing)

- Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum
- Reduce problem to non-negative spectrum by chaining

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

Constant Runtime of Single-Path Loops (Preprocessing)

- Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum
- Reduce problem to non-negative spectrum by chaining

$$\text{while } (\varphi(\mathbf{x})) \text{ do } \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$$

has constant runtime

if and only if

$$\text{while } (\varphi(\mathbf{x}) \;\wedge\; \varphi(A\mathbf{x} + \mathbf{b})) \text{ do } \quad \mathbf{x} \leftarrow A(A\mathbf{x} + \mathbf{b}) + \mathbf{b}$$

has constant runtime

Constant Runtime of Single-Path Loops (Preprocessing)

- Constant runtime is decidable over $\mathbb{R}$ if A has real spectrum
- Reduce problem to non-negative spectrum by chaining

$$\text{while } (\, \varphi(\mathbf{x}) \,) \text{ do } \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$$ has constant runtime

if and only if

$$\text{while } (\, \varphi(\mathbf{x}) \ \wedge \ \varphi(A\mathbf{x} + \mathbf{b}) \,) \text{ do } \quad \mathbf{x} \leftarrow A(A\mathbf{x} + \mathbf{b}) + \mathbf{b}$$ has constant runtime

$$\lambda_1, \ldots, \lambda_k \text{ eigenvalues of } A \quad \Longrightarrow \quad \lambda_1^2, \ldots, \lambda_k^2 \text{ eigenvalues of } A^2$$

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

  $cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n$ with polynomials α_i over $\mathbf{x}$, $a_i \in \mathbb{N}$, and $\lambda_i \in \mathbb{R}_{>0}$

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

Example:

$$\text{while } (0 \leq x+y \leq 10) \text{ do } (x, y) \leftarrow (x+1, 2 \cdot y)$$

$$cf_x(n, \mathbf{x}) = x + n \qquad \text{and} \qquad cf_y(n, \mathbf{x}) = 2^n \cdot y$$

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$\text{while } (\varphi(\mathbf{x})) \text{ do } \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

if and only if

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Example:

$\text{while } (0 \leq x + y \leq 10) \text{ do } (x, y) \leftarrow (x + 1, 2 \cdot y)$

$$cf_x(n, \mathbf{x}) = x + n \qquad \text{and} \qquad cf_y(n, \mathbf{x}) = 2^n \cdot y$$

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

  $cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n$ with polynomials α_i over $\mathbf{x}$, $a_i \in \mathbb{N}$, and $\lambda_i \in \mathbb{R}_{>0}$

  there exists $C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$

Properties of $cf(n, \mathbf{x})$:

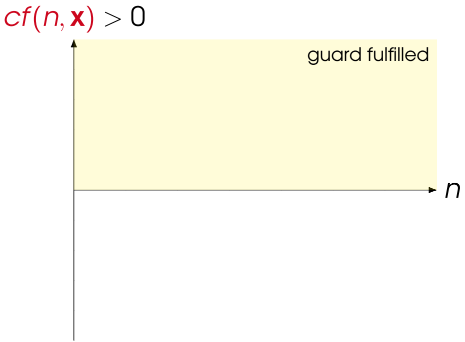
Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

  $cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n$ with polynomials α_i over $\mathbf{x}$, $a_i \in \mathbb{N}$, and $\lambda_i \in \mathbb{R}_{>0}$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:



Constant Runtime of Single-Path Loops

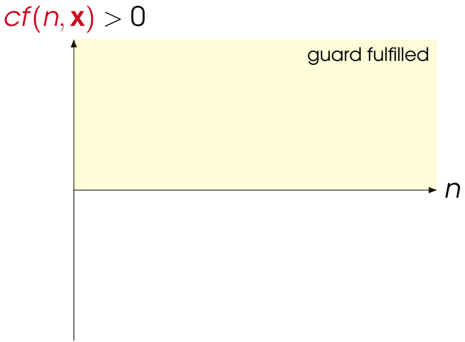
- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)



Constant Runtime of Single-Path Loops

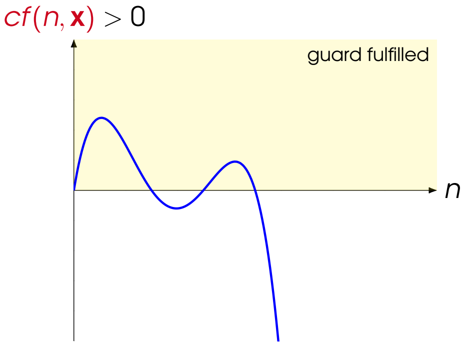
- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)



Constant Runtime of Single-Path Loops

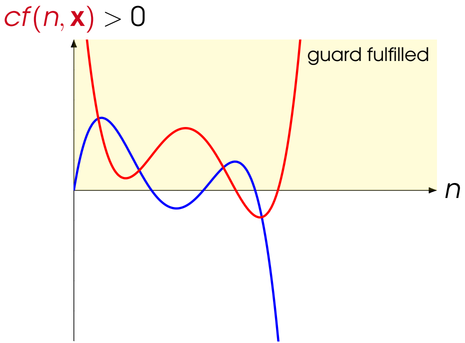
- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)



Constant Runtime of Single-Path Loops

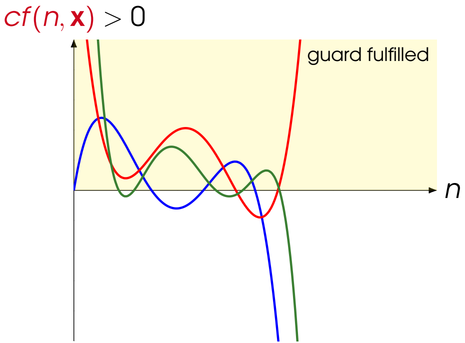
- Closed-form expression after n iterations of A:

  $cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n$ with polynomials α_i over $\mathbf{x}$, $a_i \in \mathbb{N}$, and $\lambda_i \in \mathbb{R}_{>0}$

  $$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)



Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

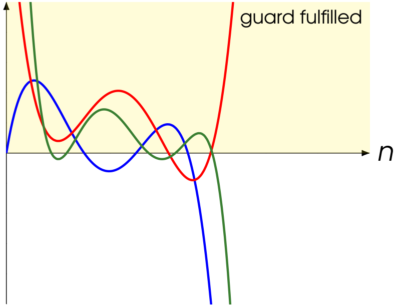
$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \ldots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)
  - Non-Termination: there exists $\mathbf{x}$ such that $cf(n, \mathbf{x})$ diverges towards ∞

$cf(n, \mathbf{x}) > 0$



Constant Runtime of Single-Path Loops

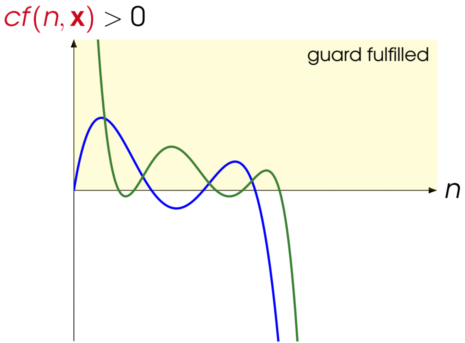
- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \dots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)
  - Non-Termination: there exists $\mathbf{x}$ such that $cf(n, \mathbf{x})$ diverges towards ∞



Constant Runtime of Single-Path Loops

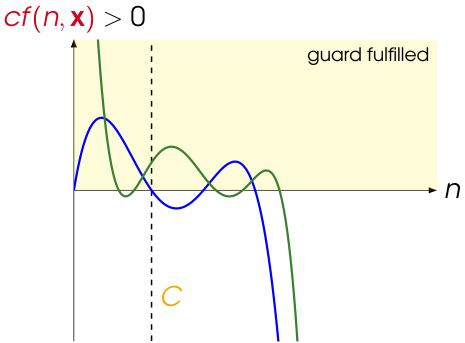
- Closed-form expression after n iterations of A:

$$cf(n,\mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0,\dots,C-1\}. \quad \neg\varphi(cf(n,\mathbf{x}))$$

Properties of $cf(n,\mathbf{x})$:

- $cf(n,\mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)
  - Non-Termination: there exists $\mathbf{x}$ such that $cf(n,\mathbf{x})$ diverges towards ∞



Constant Runtime of Single-Path Loops

- Closed-form expression after n iterations of A:

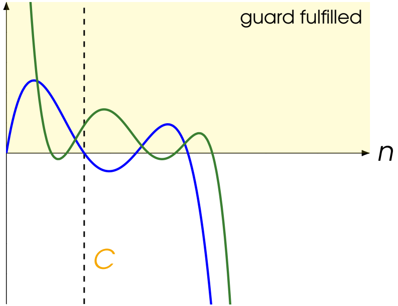
$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \dots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)
  - Non-Termination: there exists $\mathbf{x}$ such that $cf(n, \mathbf{x})$ diverges towards ∞
- $cf(n, \mathbf{x})$ has constantly many real "roots" independent of $\mathbf{x}$

$cf(n, \mathbf{x}) > 0$



Constant Runtime of Single-Path Loops

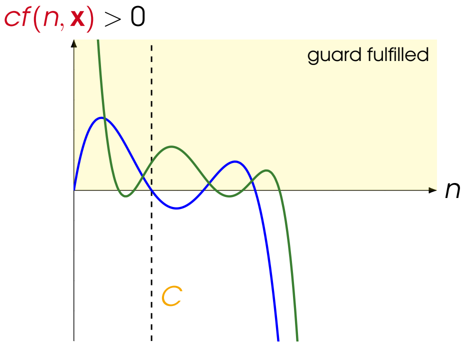
- Closed-form expression after n iterations of A:

$$cf(n, \mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) \cdot n^{a_i} \cdot \lambda_i^n \text{ with polynomials } \alpha_i \text{ over } \mathbf{x},\ a_i \in \mathbb{N}, \text{ and } \lambda_i \in \mathbb{R}_{>0}$$

$$\text{there exists } C \in \mathbb{N}. \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad \exists n \in \{0, \dots, C-1\}. \quad \neg\varphi(cf(n, \mathbf{x}))$$

Properties of $cf(n, \mathbf{x})$:

- $cf(n, \mathbf{x})$ always diverges towards $\pm\infty$ (or is constant)
  - Non-Termination: there exists $\mathbf{x}$ such that $cf(n, \mathbf{x})$ diverges towards ∞
- $cf(n, \mathbf{x})$ has constantly many real "roots" independent of $\mathbf{x}$
  - there exists constant $\mathrm{rb} \geq \#roots$



Constant Runtime of Single-Path Loops (Root Bound)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

Example:

while $(0 \leq x+y \leq 10)$ do $(x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

Constant Runtime of Single-Path Loops (Root Bound)

$\texttt{while}\,(\varphi(\mathbf{x}))\,\texttt{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \exists n \in \{0, \dots, C-1\}.\quad \neg\varphi(cf(n, \mathbf{x}))$

Example:

$\texttt{while}\,(0 \le x + y \le 10)\,\texttt{do}\,(x, y) \leftarrow (x + 1, 2 \cdot y)$ has constant runtime

Constant Runtime of Single-Path Loops (Root Bound)

$\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \exists n \in \{0, \ldots, C-1\}.\quad \neg\varphi(cf(n, \mathbf{x}))$

Example:

$\text{while}\ (0 \leq x+y \leq 10)\ \text{do}\ (x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

$\iff \exists C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \exists n \in \{0, \ldots, C-1\}.\quad \neg(0 \leq x + n + 2^n \cdot y \leq 10)$

Constant Runtime of Single-Path Loops (Root Bound)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\exists n \in \{0, \ldots, C-1\}$. $\neg\varphi(cf(n, \mathbf{x}))$

$\iff$ there exists $C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\bigvee_{n \in \{0,\ldots,\mathrm{rb}\}} \neg\varphi(cf(n \cdot C, \mathbf{x}))$

Example:

while $(0 \le x+y \le 10)$ do $(x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

$\iff$ $\exists C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\exists n \in \{0, \ldots, C-1\}$. $\neg(0 \le x + n + 2^n \cdot y \le 10)$

Constant Runtime of Single-Path Loops (Root Bound)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\exists n \in \{0, \ldots, C-1\}$. $\neg\varphi(cf(n, \mathbf{x}))$

$\iff$ there exists $C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\bigvee_{n \in \{0,\ldots,\mathrm{rb}\}} \neg\varphi(cf(n \cdot C, \mathbf{x}))$

$\iff$ there exists $C \in \mathbb{N}$. $\neg\exists \mathbf{x} \in \mathbb{R}^d$. $\bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$

Example:

while $(0 \le x + y \le 10)$ do $(x, y) \leftarrow (x + 1, 2 \cdot y)$ has constant runtime

$\iff$ $\exists C \in \mathbb{N}$. $\forall \mathbf{x} \in \mathbb{R}^d$. $\exists n \in \{0, \ldots, C-1\}$. $\neg(0 \le x + n + 2^n \cdot y \le 10)$

Constant Runtime of Single-Path Loops (Root Bound)

$\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \text{there exists } C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \exists n \in \{0, \ldots, C-1\}.\quad \neg\varphi(cf(n, \mathbf{x}))$$

$$\iff \text{there exists } C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \bigvee_{n \in \{0,\ldots,\text{rb}\}} \neg\varphi(cf(n \cdot C, \mathbf{x}))$$

$$\iff \text{there exists } C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\quad \bigwedge_{n \in \{0,\ldots,\text{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$$

Example:

$\text{while}\ (0 \le x+y \le 10)\ \text{do}\ (x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

$$\iff \exists C \in \mathbb{N}.\quad \forall \mathbf{x} \in \mathbb{R}^d.\quad \exists n \in \{0, \ldots, C-1\}.\quad \neg(0 \le x + n + 2^n \cdot y \le 10)$$

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\quad \bigwedge_{n \in \{0,\ldots,6\}} 0 \le x + n \cdot C + 2^{n \cdot C} \cdot y \le 10$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \quad \text{there exists } C \in \mathbb{N}. \quad \neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\dots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \quad \text{there exists } C \in \mathbb{N}. \quad \underbrace{\neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))}_{\text{Fourier-Motzkin elimination}}$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \text{there exists } C \in \mathbb{N}. \quad \underbrace{\neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))}_{\text{Fourier-Motzkin elimination}}$$

Example:

while $(0 \le x + y \le 10)$ do $(x, y) \leftarrow (x + 1, 2 \cdot y)$ has constant runtime

$$\iff \exists C \in \mathbb{N}. \quad \neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,6\}} 0 \le x + n \cdot C + 2^{n \cdot C} \cdot y \le 10$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

$\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \quad \text{there exists } C \in \mathbb{N}.\quad \underbrace{\neg\exists\, \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,\mathrm{rb}\}} \varphi(cf(n\cdot C, \mathbf{x}))}_{\text{Fourier-Motzkin elimination}}$$

Example:

$\text{while}\ (0 \le x+y \le 10)\ \text{do}\ (x, y) \leftarrow (x+1, 2\cdot y)$ has constant runtime

$$\iff \quad \exists\, C \in \mathbb{N}.\quad \neg\exists\, \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,6\}} 0 \le x + n\cdot C + 2^{n\cdot C}\cdot y \le 10$$

$$\iff \quad \exists\, C \in \mathbb{N}.\quad \neg\exists\, \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,6\}} -n\cdot C - 2^{n\cdot C}\cdot y \le x \le 10 - n\cdot C - 2^{n\cdot C}\cdot y$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \text{there exists } C \in \mathbb{N}.\ \underbrace{\neg \exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\dots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))}_{\text{Fourier-Motzkin elimination}}$$

Example:

while $(0 \leq x+y \leq 10)$ do $(x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

$$\iff \exists C \in \mathbb{N}.\ \neg \exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\dots,6\}} 0 \leq x + n \cdot C + 2^{n \cdot C} \cdot y \leq 10$$

$$\iff \exists C \in \mathbb{N}.\ \neg \exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\dots,6\}} -n \cdot C - 2^{n \cdot C} \cdot y \leq x \leq 10 - n \cdot C - 2^{n \cdot C} \cdot y$$

$$\iff \exists C \in \mathbb{N}.\ \neg(0 \leq -4^C \cdot C + 10 \cdot 4^C + 2^C \cdot 2 \cdot C + 10 \cdot 2^C - C - 20 \wedge \dots)$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \text{there exists } C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$$

Example:

while $(0 \leq x+y \leq 10)$ do $(x, y) \leftarrow (x+1, 2 \cdot y)$ has constant runtime

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\ldots,6\}} 0 \leq x + n \cdot C + 2^{n \cdot C} \cdot y \leq 10$$

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n \in \{0,\ldots,6\}} -n \cdot C - 2^{n \cdot C} \cdot y \leq x \leq 10 - n \cdot C - 2^{n \cdot C} \cdot y$$

$$\iff \exists C \in \mathbb{N}.\quad \neg(0 \leq -4^C \cdot C + 10 \cdot 4^C + 2^C \cdot 2 \cdot C + 10 \cdot 2^C - C - 20 \wedge \ldots)$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

$\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$$\iff \text{there exists } C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$$

$$\iff \text{there exists } C \in \mathbb{N}.\quad \neg \bigwedge 0 \underset{(-)}{\leq} pe$$ (pe poly-exp expression over C)

Example:

$\text{while}\ (0 \leq x + y \leq 10)\ \text{do}\ (x, y) \leftarrow (x + 1, 2 \cdot y)$ has constant runtime

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,6\}} 0 \leq x + n \cdot C + 2^{n \cdot C} \cdot y \leq 10$$

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,6\}} -n \cdot C - 2^{n \cdot C} \cdot y \leq x \leq 10 - n \cdot C - 2^{n \cdot C} \cdot y$$

$$\iff \exists C \in \mathbb{N}.\quad \neg(0 \leq -4^C \cdot C + 10 \cdot 4^C + 2^C \cdot 2 \cdot C + 10 \cdot 2^C - C - 20 \wedge \ldots)$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

$\texttt{while}\ (\varphi(\mathbf{x}))\ \texttt{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$

$\iff$ there exists $C \in \mathbb{N}.\quad \neg \bigwedge 0 \underset{(-)}{<} pe$ (pe poly-exp expression over C)

Example:

$\texttt{while}\ (0 \le x+y \le 10)\ \texttt{do}\ (x, y) \leftarrow (x+1, 2\cdot y)$ has constant runtime $C \cdot \mathrm{rb} = 66$

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,6\}} 0 \le x + n \cdot C + 2^{n\cdot C} \cdot y \le 10$$

$$\iff \exists C \in \mathbb{N}.\quad \neg\exists \mathbf{x} \in \mathbb{R}^d.\ \bigwedge_{n\in\{0,\ldots,6\}} -n \cdot C - 2^{n\cdot C} \cdot y \le x \le 10 - n \cdot C - 2^{n\cdot C} \cdot y$$

$$\iff \exists C \in \mathbb{N}.\quad \neg(0 \le -4^C \cdot C + 10 \cdot 4^C + 2^C \cdot 2 \cdot C + 10 \cdot 2^C - C - 20 \wedge \ldots)$$

Constant Runtime of Single-Path Loops (Fourier-Motzkin)

while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$ has constant runtime

$\iff$ there exists $C \in \mathbb{N}$. $\neg \exists \mathbf{x} \in \mathbb{R}^d. \bigwedge_{n \in \{0,\ldots,\mathrm{rb}\}} \varphi(cf(n \cdot C, \mathbf{x}))$

$\iff$ there exists $C \in \mathbb{N}$. $\neg \bigwedge 0 \underset{(-)}{<} pe$ (pe poly-exp expression over C)

Decidability Results:

Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum

Loopy

Implemented in Loopy using SymPy:

Loopy

Implemented in Loopy using SymPy:

- Experiments on linear single-path loops from the TPDB (1336 examples)

Loopy

Implemented in Loopy using SymPy:

- Experiments on linear single-path loops from the TPDB (1336 examples)
- Loopy average runtime of 1.71 seconds

Tool	$\mathcal{O}(1)$	$\neq \mathcal{O}(1)$
Loopy	76	1260

Loopy

Implemented in Loopy using SymPy:

- Experiments on linear single-path loops from the TPDB (1336 examples)
- Loopy average runtime of 1.71 seconds

Comparison to complexity analysis tools

- lower bounds: LoAT
- upper bounds: CoFloCo and KoAT

Tool	$\mathcal{O}(1)$	$\neq \mathcal{O}(1)$
Loopy	76	1260
KoAT	6	—
CoFloCo	5	—
LoAT	—	1226

Conclusion

Constant runtime of loops like $\text{while}\,(\,\varphi(\mathbf{x})\,)\,\text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Conclusion

Constant runtime of loops like $\text{while} \; (\varphi(\mathbf{x})) \; \text{do} \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

Constant runtime of loops like $\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum

Conclusion

Constant runtime of loops like while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum

Conclusion

Constant runtime of loops like while $(\varphi(\mathbf{x}))$ do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Conclusion

Constant runtime of loops like $\text{while } (\varphi(\mathbf{x})) \text{ do } \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Future Work:

Conclusion

Constant runtime of loops like $\text{while } (\varphi(\mathbf{x})) \text{ do } \quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Future Work:

- Extend approach over real spectrum by polyhedral preconditions

Conclusion

Constant runtime of loops like $\text{while}\ (\varphi(\mathbf{x}))\ \text{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Future Work:

- Extend approach over real spectrum by polyhedral preconditions
- Handle arbitrary complex-valued eigenvalues

Conclusion

Constant runtime of loops like $\texttt{while}\ (\varphi(\mathbf{x}))\ \texttt{do}\quad \mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Future Work:

- Extend approach over real spectrum by polyhedral preconditions
- Handle arbitrary complex-valued eigenvalues

Implementation in Loopy

Conclusion

Constant runtime of loops like   while ($\varphi(\mathbf{x})$) do $\mathbf{x} \leftarrow A\mathbf{x} + \mathbf{b}$

Decidability Results:

- Constant runtime is **decidable** over $\mathbb{R}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Q}$ if A has real spectrum
- Constant runtime is **decidable** over $\mathbb{Z}$ if A has spectrum $\{-1, 0, 1\}$

Future Work:

- Extend approach over real spectrum by polyhedral preconditions
- Handle arbitrary complex-valued eigenvalues

Implementation in Loopy

Thank You!