What Else is Undecidable about Loops?

Laura Kovács and Anton Varonka

TU WIEN Informatics*

RAMiCS 2023

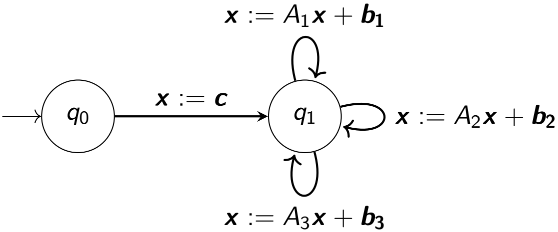
Augsburg, April 3-6

*Supported by the ERC CoG ARTIST 101002685, ProbInG project of the Vienna Science and Technology Fund, and the Marie Skłodowska-Curie Network LogiCS@TU Wien.

A simple loop acting on a vector $\boldsymbol{x}$ of integer variables.

Program correctness:

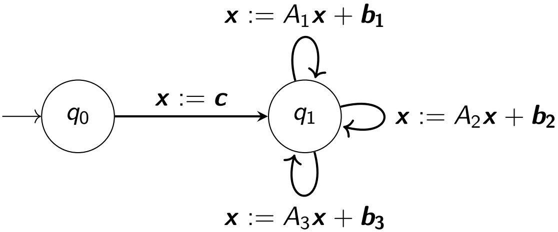
- Termination on all branches
- Finding good invariants



A simple loop acting on a vector $\boldsymbol{x}$ of integer variables.

Program correctness:

- Termination on all branches
- Finding good invariants



The Collatz Problem

$x :=$ input()
while $x \neq 1$ **do**
  $x \bmod 2 = 0 \rightarrow x := \frac{1}{2}x$
  $x \bmod 2 = 1 \rightarrow x := 3x + 1$

any input from the domain $\mathbb{N}$
does program terminate on all of
them?

The Collatz Problem

$$
\begin{aligned}
&x := \text{input}() \\
&\textbf{while } x \neq 1 \textbf{ do} \\
&\quad x \bmod 2 = 0 \rightarrow x := \tfrac{1}{2}x \\
&\quad x \bmod 2 = 1 \rightarrow x := 3x + 1
\end{aligned}
$$

any input from the domain $\mathbb{N}$
does program terminate on all of them?

This is yet not an undecidable problem about loops.

Collatz as Decision Problem

$$
\begin{array}{l}
x := c \\
\textbf{while } x \neq 1 \textbf{ do} \\
\quad x \bmod N = 0 \rightarrow x := a_0 x + b_0 \\
\qquad \vdots \\
\quad x \bmod N = N-1 \rightarrow x := a_{N-1} x + b_{N-1}
\end{array}
$$

Given N and rational $a_0, b_0, \ldots, a_{N-1}, b_{N-1}$. Does the loop terminate for all $c \in \mathbb{N}$?

Collatz as Decision Problem

$x := c$
while $x \neq 1$ **do**
  $x \bmod N = 0 \to x := a_0 x + b_0$
  $\vdots$
  $x \bmod N = N-1 \to x := a_{N-1} x + b_{N-1}$

Given N and rational $a_0, b_0, \ldots, a_{N-1}, b_{N-1}$. Does the loop terminate for all $c \in \mathbb{N}$?

Generalised Collatz is

undecidable (Conway + Kurtz & Simon)

Collatz as Decision Problem

$$
\begin{array}{l}
x := c \\
\textbf{while } x \neq 1 \textbf{ do} \\
\quad x \bmod N = 0 \to x := a_0 x + b_0 \\
\quad\ \vdots \\
\quad x \bmod N = N-1 \to x := a_{N-1}x + b_{N-1}
\end{array}
$$

Given N and rational $a_0, b_0, \ldots, a_{N-1}, b_{N-1}$. Does the loop terminate for all $c \in \mathbb{N}$?

Generalised Collatz is

undecidable (Conway + Kurtz & Simon)

and so is the termination of piecewise affine loops.

Single Update Termination

$$
\begin{array}{l}
\boldsymbol{x} := \boldsymbol{c} \\
\textbf{while } x_1 \geq 0 \textbf{ do} \\
\quad \boldsymbol{x} := A \cdot \boldsymbol{x}
\end{array}
$$

Here: $\boldsymbol{x} = \begin{bmatrix} x_1 \dots x_d \end{bmatrix}^T \in \mathbb{Z}^d$
and A is a linear transformation of $\mathbb{Z}^d$.
Does the loop terminate for all $\boldsymbol{c} \in \mathbb{Z}^d$?

Theorem [Hosseini, Ouaknine, Worrell (ICALP'19)]

Universal Termination of single-path linear loops over $\mathbb{Z}$ is decidable.

Which of the nuances made the difference?

- universal termination: both problems
- updates are linear (affine): both problems
- number of variables: one vs many
- guards: linear equalities vs linear inequalities

Which of the nuances made the difference?

- universal termination: both problems
- updates are linear (affine): both problems
- number of variables: one vs many
- guards: linear equalities vs linear inequalities
- conditional branching?

Mark Braverman
(2022 IMU Abacus Medal)

Q: "How much non-determinism can be introduced in a linear loop [. . .] before termination becomes undecidable?" (Braverman, 2006)

Programs we consider

Non-deterministic.

Sample loop

while **do**

σ_1 :

or

σ_2 :

or

σ_3 :

Programs we consider

Non-deterministic. Arbitrary dimension.

Sample loop (in 3D)

$(x, y, z) := (-1, -1, 2)$

while **do**

$\boldsymbol{\sigma_1}$: $(x, y, z) := (x - y + 1, y - 2z, 2z - x - 1)$

or

$\boldsymbol{\sigma_2}$: $(x, y, z) := (-\frac{3}{2}, x + y + \frac{1}{2}, -x - y + 1)$

or

$\boldsymbol{\sigma_3}$: $(x, y, z) := (2x, y + z, -x)$

Programs we consider

Non-deterministic. Arbitrary dimension. Linear inequality conditions.

Sample loop (in 3D)

$(x, y, z) := (-1, -1, 2)$
while $x + 2y + 3z > 0 \wedge x \leq 10$ **do**
$\boldsymbol{\sigma_1}$: $(x, y, z) := (x - y + 1, y - 2z, 2z - x - 1)$
or
$\boldsymbol{\sigma_2}$: $(x, y, z) := (-\frac{3}{2}, x + y + \frac{1}{2}, -x - y + 1)$
or
$\boldsymbol{\sigma_3}$: $(x, y, z) := (2x, y + z, -x)$

Non-determinism interpreted

Fix an input s. Termination from s — no infinite executions.

Non-determinism interpreted

Fix an input s. Termination from s — no infinite executions.
Determinism: one execution per input;

130

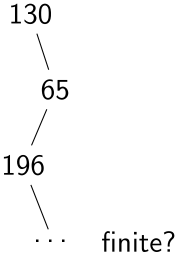
65

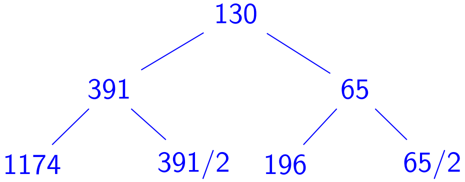
196

$\cdots$ finite?

Non-determinism interpreted

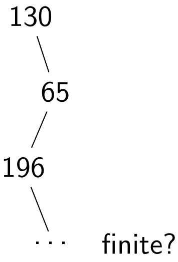
Fix an input s. Termination from s — no infinite executions.
Determinism: one execution per input; with non-determinism: branching

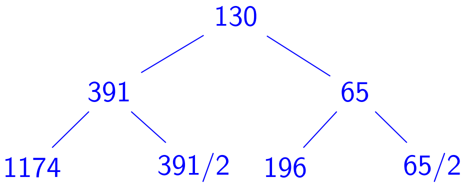




Non-determinism interpreted

Fix an input s. Termination from s — no infinite executions.
Determinism: one execution per input; with non-determinism: branching

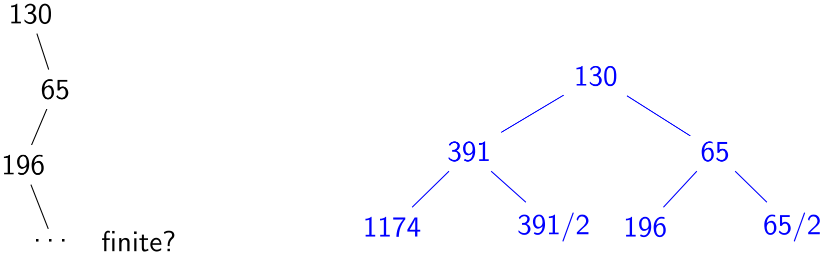




Termination is no longer just reachability.

Non-determinism interpreted

Fix an input s. Termination from s — no infinite executions.
Determinism: one execution per input; with non-determinism: branching



Termination is no longer just reachability.

Motivation from program correctness: terminating on all branches.

Termination Problem

Termination (on a set)

Given: a loop $\mathcal{L}$ and a set of inputs $S \subseteq \mathbb{Z}^d$.
Does $\mathcal{L}$ terminate on every input from S?

Termination Problem

Termination (on a set)

Given: a loop $\mathcal{L}$ and a set of inputs $S \subseteq \mathbb{Z}^d$.
Does $\mathcal{L}$ terminate on every input from S?

The Halting Problem: S is a singleton.
The Universal Termination Problem: $S = \mathbb{Z}^d$.

Termination: the result

Theorem

Termination of multi-path affine loops with linear inequality conditions **is undecidable**.

Proof by reduction from the Post's Correspondence Problem (its complement). A loop terminates on a set S iff an instance of PCP has no solution.

Remains undecidable with:

- just 4 variables, or
- just 2 linear updates.

Termination undecidable

PCP input: $\{\frac{011}{0}\}, \{\frac{1}{11}\}$.

$$\frac{1}{1} \to \frac{1011}{10} \to \frac{10111}{1011} \to \frac{101111}{101111} \qquad \mapsto \qquad \frac{1}{1} \xrightarrow{\sigma_1} \frac{11}{2} \xrightarrow{\sigma_2} \frac{23}{11} \xrightarrow{\sigma_2} \frac{47}{47}$$

$(x, y, z) := (1, 1, 1)$
while $c \geq 0 \land z \geq 0 \land z \leq 1$ **do**
  $\boldsymbol{\sigma_1}$ or $\boldsymbol{\sigma_2}$ or $\boldsymbol{\sigma_3}$

Updates σ_1 and σ_2 guarantee: a fixpoint $\begin{pmatrix}47 & 47 & 0 & 0\end{pmatrix}$ of σ_3 is reached
$\sigma_1\sigma_2\sigma_2(\sigma_3)^\omega$ is a non-terminating execution
other executions: forced to apply σ_1 or σ_2 until $x = y$, otherwise termination in at most 2 steps

Half-time

Source: FC Augsburg on Twitter, 01/08/2015

Half-time

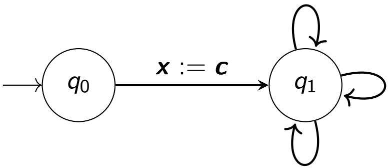
Questions so far?

Source: FC Augsburg on Twitter, 01/08/2015

A simple loop acting on a vector $\boldsymbol{x}$ of integer variables.

Program correctness:

- Termination on all branches
- Finding good invariants



Invariants

$$
\begin{array}{l}
(x, y, z) := (-1, -1, 2) \\
\textbf{while } \text{true } \textbf{do} \\
\quad (x, y, z) := (x - y + 1, y - z, 2z + y - 1) \\
\quad \text{or} \\
\quad (x, y, z) := (-\frac{3}{2}, x + y + \frac{1}{2}, z + 1) \\
\quad \text{or} \\
\quad (x, y, z) := (2x, y + z, -x)
\end{array}
$$

Inductive invariant is a relation between variables of a loop $\mathcal{L}$ which is *preserved under any update of* $\mathcal{L}$:

$$f(x, y, z) = 0.$$

Invariants

$(x, y, z) := (-1, -1, 2)$
while true **do**
  $(x, y, z) := (x - y + 1, y - z, 2z + y - 1)$
  or
  $(x, y, z) := (-\frac{3}{2}, x + y + \frac{1}{2}, z + 1)$
  or
  $(x, y, z) := (2x, y + z, -x)$

Inductive invariant is a relation between variables of a loop $\mathcal{L}$ which is *preserved under any update of* $\mathcal{L}$:

$$x + y + z = 0.$$

Algebraic invariants

Algebraic invariants are those of the form

$$p(x_1, \dots, x_d) = 0,$$

where $p \in \mathbb{Q}[x_1, \dots, x_d]$ is a multivariate polynomial.

Algebraic invariants

Algebraic invariants are those of the form

$$p(x_1, \ldots, x_d) = 0,$$

where $p \in \mathbb{Q}[x_1, \ldots, x_d]$ is a multivariate polynomial.

Theorem [Müller-Olm, Seidl (2004)]

There is an algorithm to compute all <u>algebraic invariants of given degree d</u> for programs with polynomial updates.

Algebraic invariants

Algebraic invariants are those of the form

$$p(x_1, \dots, x_d) = 0,$$

where $p \in \mathbb{Q}[x_1, \dots, x_d]$ is a multivariate polynomial.

Theorem [Müller-Olm, Seidl (2004)]

There is an algorithm to compute all algebraic invariants of given degree d for programs with polynomial updates.

All algebraic invariants of a loop build a polynomial ideal.
It can be finitely represented:

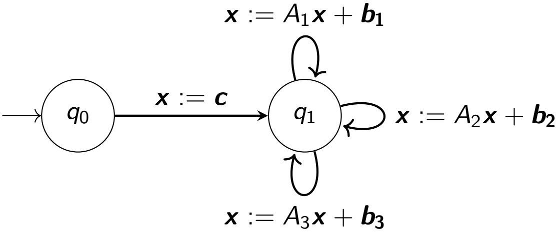
There exists the strongest algebraic invariant.

$$x^2 - y^3 = 0 \wedge y - 2z + 1 = 0$$

Multi-Path Affine Loops

Theorem [Hrushovski et al. (LICS'18)]

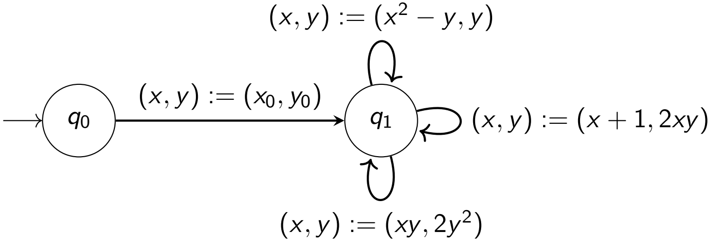
There exists an algorithm to compute the strongest algebraic invariant of a multi-path affine loop.



Invariants beyond Affine Loops

Theorem

Finding the strongest algebraic invariant of a multi-path loop with update degrees ≤ 2 is algorithmically **unsolvable**.

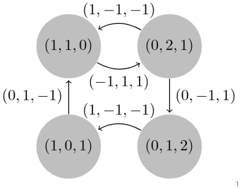


Invariants Unsolvability

Theorem

Finding the strongest algebraic invariant of a multi-path loop with update degrees ≤ 2 is algorithmically **unsolvable**.

Proof idea: reduction from the undecidable *Reset VASS* Boundedness.



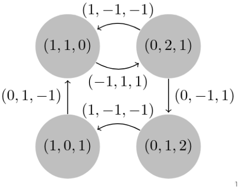
VASS. Source: Jérôme Leroux, arXiv.org

In VASS, all valuations are non-negative.

Theorem

Finding the strongest algebraic invariant of a multi-path loop with update degrees ≤ 2 is algorithmically **unsolvable**.

Proof idea: reduction from the undecidable *Reset VASS* Boundedness.



VASS. Source: Jérôme Leroux, arXiv.org

Boundedness: is the set of reachable valuations in q finite?

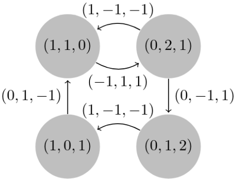
In VASS, all valuations are non-negative.

Invariants Unsolvability

Theorem

Finding the strongest algebraic invariant of a multi-path loop with update degrees ≤ 2 is algorithmically **unsolvable**.

Proof idea: reduction from the undecidable *Reset VASS* Boundedness.



VASS. Source: Jérôme Leroux, arXiv.org

Boundedness: is the set of reachable valuations in q finite?

In a loop, blocked transitions are simulated with updates of degree 2.

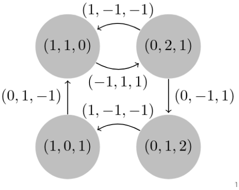
In VASS, all valuations are non-negative.

Invariants Unsolvability

Theorem

Finding the strongest algebraic invariant of a multi-path loop with update degrees ≤ 2 is algorithmically **unsolvable**.

Proof idea: reduction from the undecidable *Reset VASS* Boundedness.



VASS. Source: Jérôme Leroux, arXiv.org

In VASS, all valuations are non-negative.

Boundedness: is the set of reachable valuations in q finite?

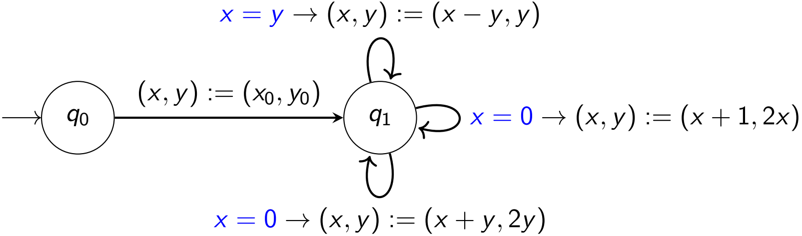
In a loop, blocked transitions are simulated with updates of degree 2.

VASS bounded iff in a multi-path loop, the strongest algebraic invariant has dimension ≤ 1.

What Else is Unsolvable?

Proposition

Finding the strongest algebraic invariant of a multi-path affine loop with guarded affine updates is algorithmically **unsolvable**.



Open questions

1. The Halting Problem for multi-path affine loops;

Open questions

1. The Halting Problem for multi-path affine loops;
2. The Strongest Algebraic Invariant for deterministic loops with non-affine updates

$$\textit{while}\ \ \mathsf{true}\ \ \textit{do}\ \ (x, y) := (x^2 - y, xy);$$

Open questions

1. The Halting Problem for multi-path affine loops;
2. The Strongest Algebraic Invariant for deterministic loops with non-affine updates

$$\textit{while}\ \ \text{true}\ \ \textit{do}\ \ (x, y) := (x^2 - y, xy);$$

3. The Termination Problem for linear-constraint loops

$$\textit{while}\ \ B\boldsymbol{x} \geq \boldsymbol{b}\ \ \textit{do}\ \ A\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x'}\end{bmatrix}^T \leq \boldsymbol{c}.$$

Take-away

Undecidable termination:

non-determinism +
affine updates +
linear inequality conditions

Unsolvable invariant generation:

non-determinism +
quadratic updates *or*
affine equality guards

Thank You!