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

$$\textbf{while } \varphi \textbf{ do } \overline{x} \ \leftarrow A\,\overline{x} + \overline{a}. \tag{1}$$

Here, $A \in \mathbb{Z}^{d \times d}$ for some dimension $d \geq 1$, $\overline{x}$ is a column vector of pairwise different variables $x_1, \ldots, x_d$, $\overline{a} \in \mathbb{Z}^d$, and φ is a conjunction of inequalities of the form $\alpha > 0$ where $\alpha \in \mathbb{A}\mathbb{f}[\overline{x}]$ is an affine expression with rational coefficients[1] over $\overline{x}$ (i.e., $\mathbb{A}\mathbb{f}[\overline{x}] = \{\overline{c}^T\,\overline{x} + c \mid \overline{c} \in \mathbb{Q}^d, c \in \mathbb{Q}\}$). So φ has the form $B\,\overline{x} + \overline{b} > \overline{0}$ where $\overline{0}$ is the vector containing k zeros, $B \in \mathbb{Q}^{k \times d}$, and $\overline{b} \in \mathbb{Q}^k$ for some $k \in \mathbb{N}$. Definition 1 formalizes the intuitive notion of termination for such loops.

Definition 1 (Termination). *Let $f : \mathbb{Z}^d \to \mathbb{Z}^d$ with $f(\overline{x}) = A\,\overline{x} + \overline{a}$. If*

$$\exists \overline{c} \in \mathbb{Z}^d.\ \forall n \in \mathbb{N}.\ \varphi[\overline{x}/f^n(\overline{c})],$$

then (1) is non-terminating *and $\overline{c}$ is a* witness *for non-termination. Otherwise, (1)* terminates.

Here, f^n denotes the n-fold application of f, i.e., we have $f^0(\overline{c}) = \overline{c}$ and $f^{n+1}(\overline{c}) = f(f^n(\overline{c}))$. We call f the *update* of (1). Moreover, for any entity s, $s[x/t]$ denotes the entity that results from s by replacing all occurrences of x by t. Similarly, if $\overline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$ and $\overline{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_m \end{bmatrix}$, then $s[\overline{x}/\overline{t}]$ denotes the entity resulting from s by replacing all occurrences of x_i by t_i for each $1 \leq i \leq m$.

[1] Note that multiplying with the least common multiple of all denominators yields an equivalent constraint with integer coefficients, i.e., allowing rational instead of integer coefficients does not extend the considered class of loops.

Funded by DFG grant 389792660 as part of TRR 248 and by DFG grant GI 274/6.

I. Dillig and S. Tasiran (Eds.): CAV 2019, LNCS 11562, pp. 426–444, 2019.
https://doi.org/10.1007/978-3-030-25543-5_24

Example 2. *Consider the loop*

$$\textbf{while } y + z > 0 \textbf{ do } \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \leftarrow \begin{bmatrix} 2 \\ x + 1 \\ -w - 2 \cdot y \\ x \end{bmatrix}$$

where the update of all variables is executed simultaneously. This program belongs to our class of affine loops, because it can be written equivalently as follows.

$$\textbf{while } y + z > 0 \textbf{ do } \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

While termination of affine loops is known to be decidable if the variables range over the real [15] or the rational numbers [4], the integer case is a well-known open problem [2–4,14,15].[2] However, certain special cases have been solved: Braverman [4] showed that termination of *linear* loops is decidable (i.e., loops of the form (1) where $\overline{a}$ is $\overline{0}$ and φ is of the form $B\,\overline{x} > \overline{0}$). Bozga et al. [3] showed decidability for the case that the update matrix A in (1) has the *finite monoid property*, i.e., if there is an $n > 0$ such that A^n is diagonalizable and all eigenvalues of A^n are in $\{0, 1\}$. Ouaknine et al. [14] proved decidability for the case $d \leq 4$ and for the case that A is diagonalizable.

Ben-Amram et al. [2] showed undecidability of termination for certain extensions of affine integer loops, e.g., for loops where the body is of the form **if** $x > 0$ **then** $\overline{x} \leftarrow A\,\overline{x}$ **else** $\overline{x} \leftarrow A'\,\overline{x}$ where $A, A' \in \mathbb{Z}^{d \times d}$ and $x \in \overline{x}$.

In this paper, we present another substantial step towards the solution of the open problem whether termination of affine integer loops is decidable. We show that termination is decidable for *triangular* loops (1) where A is a triangular matrix (i.e., all entries of A below or above the main diagonal are zero). Clearly, the order of the variables is irrelevant, i.e., our results also cover the case that A can be transformed into a triangular matrix by reordering A, $\overline{x}$, and $\overline{a}$ accordingly.[3] So essentially, triangularity means that the program variables $x_1, \ldots, x_d$ can be ordered such that in each loop iteration, the new value of x_i only depends on the previous values of $x_1, \ldots, x_{i-1}, x_i$. Hence, this excludes programs with "cyclic dependencies" of variables (e.g., where the new values of x and y both depend on the old values of both x and y). While triangular loops are a very restricted subclass of general integer programs, integer programs often contain such loops. Hence, tools for termination analysis of such programs (e.g., [5–8,11–13]) could

[2] The proofs for real or rational numbers do not carry over to the integers since [15] uses Brouwer's Fixed Point Theorem which is not applicable if the variables range over $\mathbb{Z}$ and [4] relies on the density of $\mathbb{Q}$ in $\mathbb{R}$.

[3] Similarly, one could of course also use other termination-preserving pre-processings and try to transform a given program into a triangular loop.

benefit from integrating our decision procedure and applying it whenever a subprogram is an affine triangular loop.

Note that triangularity and diagonalizability of matrices do not imply each other. As we consider loops with arbitrary dimension, this means that the class of loops considered in this paper is not covered by [3,14]. Since we consider affine instead of linear loops, it is also orthogonal to [4].

To see the difference between our and previous results, note that a triangular matrix A where $c_1, \ldots, c_k$ are the *distinct* entries on the diagonal is diagonalizable iff $(A - c_1 I) \ldots (A - c_k I)$ is the zero matrix.[4] Here, I is the identity matrix. So an easy example for a triangular loop where the update matrix is not diagonalizable is the following well-known program (see, e.g., [2]):

$$\textbf{while } x > 0 \textbf{ do } x \leftarrow x + y;\ y \leftarrow y - 1$$

It terminates as y eventually becomes negative and then x decreases in each iteration. In matrix notation, the loop body is $\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, i.e., the update matrix is triangular. Thus, this program is in our class of programs where we show that termination is decidable. However, the only entry on the diagonal of the update matrix A is $c = 1$ and $A - c\,I = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ is not the zero matrix. So A (and in fact each A^n where $n \in \mathbb{N}$) is not diagonalizable. Hence, extensions of this example to a dimension greater than 4 where the loop is still triangular are not covered by any of the previous results.[5]

Our proof that termination is decidable for triangular loops proceeds in three steps. We first prove that termination of triangular loops is decidable iff termination of *non-negative triangular* loops (*nnt-loops*) is decidable, cf. Sect. 2. A loop is non-negative if the diagonal of A does not contain negative entries. Second, we show how to compute *closed forms* for nnt-loops, i.e., vectors $\overline{q}$ of d expressions over the variables $\overline{x}$ and n such that $\overline{q}[n/c] = f^c(\overline{x})$ for all $c \geq 0$, see Sect. 3. Here, triangularity of the matrix A allows us to treat the variables step by step. So for any $1 \leq i \leq d$, we already know the closed forms for $x_1, \ldots, x_{i-1}$ when computing the closed form for x_i. The idea of computing closed forms for the repeated updates of loops was inspired by our previous work on inferring lower bounds on the runtime of integer programs [10]. But in contrast to [10], here the computation of the closed form always succeeds due to the restricted shape of the programs. Finally, we explain how to decide termination of nnt-loops by reasoning about their closed forms in Sect. 4. While our technique does not yield witnesses for non-termination, we show that it yields witnesses for *eventual* non-termination, i.e., vectors $\overline{c}$ such that $f^n(\overline{c})$ witnesses non-termination for some $n \in \mathbb{N}$. Detailed proofs for all lemmas and theorems can be found in [9].

[4] The reason is that in this case, $(x - c_1) \ldots (x - c_k)$ is the minimal polynomial of A and diagonalizability is equivalent to the fact that the minimal polynomial is a product of distinct linear factors.

[5] For instance, consider **while** $x > 0$ **do** $x \leftarrow x + y + z_1 + z_2 + z_3;\ y \leftarrow y - 1$.

2 From Triangular to Non-Negative Triangular Loops

To transform triangular loops into nnt-loops, we define how to *chain* loops. Intuitively, chaining yields a new loop where a single iteration is equivalent to two iterations of the original loop. Then we show that chaining a triangular loop always yields an nnt-loop and that chaining is equivalent w.r.t. termination.

Definition 3 (Chaining). Chaining *the loop (1) yields:*

$$\textbf{while } \varphi \wedge \varphi[\overline{x}/A\,\overline{x}+\overline{a}] \textbf{ do } \overline{x} \leftarrow A^2\,\overline{x} + A\,\overline{a} + \overline{a} \tag{2}$$

Example 4. *Chaining Example 2 yields*

$$\textbf{while } y+z>0 \wedge -w-2\cdot y+x>0 \textbf{ do } \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^2 \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

which simplifies to the following nnt-loop:

$$\textbf{while } y+z>0 \wedge -w-2\cdot y+x>0 \textbf{ do } \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ -2 \\ 1 \end{bmatrix}$$

Lemma 5 is needed to prove that (2) is an nnt-loop if (1) is triangular.

Lemma 5 (Squares of Triangular Matrices). *For every triangular matrix A, A^2 is a triangular matrix whose diagonal entries are non-negative.*

Corollary 6 (Chaining Loops). *If (1) is triangular, then (2) is an nnt-loop.*

Proof. Immediate consequence of Definition 3 and Lemma 5. □

Lemma 7 (Equivalence of Chaining). *(1) terminates $\iff$ (2) terminates.*

Proof. By Definition 1, (1) does not terminate iff

$$\begin{aligned} & \exists \overline{c} \in \mathbb{Z}^d.\ \forall n \in \mathbb{N}.\ \varphi[\overline{x}/f^n(\overline{c})] \\ \iff & \exists \overline{c} \in \mathbb{Z}^d.\ \forall n \in \mathbb{N}.\ \varphi[\overline{x}/f^{2\cdot n}(\overline{c})] \wedge \varphi[\overline{x}/f^{2\cdot n+1}(\overline{c})] \\ \iff & \exists \overline{c} \in \mathbb{Z}^d.\ \forall n \in \mathbb{N}.\ \varphi[\overline{x}/f^{2\cdot n}(\overline{c})] \wedge \varphi[\overline{x}/A\,f^{2\cdot n}(\overline{c})+\overline{a}] \text{ (by Definition of } f), \end{aligned}$$

i.e., iff (2) does not terminate as $f^2(\overline{x}) = A^2\,\overline{x} + A\,\overline{a} + \overline{a}$ is the update of (2). □

Theorem 8 (Reducing Termination to nnt-Loops). *Termination of triangular loops is decidable iff termination of nnt-loops is decidable.*

Proof. Immediate consequence of Corollary 6 and Lemma 7. □

Thus, from now on we restrict our attention to nnt-loops.

3 Computing Closed Forms

The next step towards our decidability proof is to show that $f^n(\overline{x})$ is equivalent to a vector of *poly-exponential expressions* for each nnt-loop, i.e., the closed form of each nnt-loop can be represented by such expressions. Here, *equivalence* means that two expressions evaluate to the same result for all variable assignments.

Poly-exponential expressions are sums of arithmetic terms where it is always clear which addend determines the asymptotic growth of the whole expression when increasing a designated variable n. This is crucial for our decidability proof in Sect. 4. Let $\mathbb{N}_{\geq 1} = \{b \in \mathbb{N} \mid b \geq 1\}$ (and $\mathbb{Q}_{>0}$, $\mathbb{N}_{>1}$, etc. are defined analogously). Moreover, $\mathbb{A}\mathrm{f}[\overline{x}]$ is again the set of all affine expressions over $\overline{x}$.

Definition 9 (Poly-Exponential Expressions). *Let $\mathcal{C}$ be the set of all finite conjunctions over the literals $n = c, n \neq c$ where n is a designated variable and $c \in \mathbb{N}$. Moreover for each formula ψ over n, let $[\![\psi]\!]$ be the characteristic function of ψ, i.e., $[\![\psi]\!](c) = 1$ if $\psi[n/c]$ is valid and $[\![\psi]\!](c) = 0$, otherwise. The set of all* poly-exponential expressions *over $\overline{x}$ is*

$$\mathbb{PE}[\overline{x}] = \left\{ \sum_{j=1}^{\ell} [\![\psi_j]\!] \cdot \alpha_j \cdot n^{a_j} \cdot b_j^n \;\middle|\; \ell, a_j \in \mathbb{N},\ \psi_j \in \mathcal{C},\ \alpha_j \in \mathbb{A}\mathrm{f}[\overline{x}],\ b_j \in \mathbb{N}_{\geq 1} \right\}.$$

As n ranges over $\mathbb{N}$, we use $[\![n > c]\!]$ as syntactic sugar for $[\![\bigwedge_{i=0}^{c} n \neq i]\!]$. So an example for a poly-exponential expression is

$$[\![n > 2]\!] \cdot (2 \cdot x + 3 \cdot y - 1) \cdot n^3 \cdot 3^n \;+\; [\![n = 2]\!] \cdot (x - y).$$

Moreover, note that if ψ contains a *positive* literal (i.e., a literal of the form "$n = c$" for some number $c \in \mathbb{N}$), then $[\![\psi]\!]$ is equivalent to either 0 or $[\![n = c]\!]$.

The crux of the proof that poly-exponential expressions can represent closed forms is to show that certain sums over products of exponential and poly-exponential expressions can be represented by poly-exponential expressions, cf. Lemma 12. To construct these expressions, we use a variant of [1, Lemma 3.5]. As usual, $\mathbb{Q}[\overline{x}]$ is the set of all polynomials over $\overline{x}$ with rational coefficients.

Lemma 10 (Expressing Polynomials by Differences [1]). *If $q \in \mathbb{Q}[n]$ and $c \in \mathbb{Q}$, then there is an $r \in \mathbb{Q}[n]$ such that $q = r - c \cdot r[n/n-1]$ for all $n \in \mathbb{N}$.*

So Lemma 10 expresses a polynomial q via the difference of another polynomial r at the positions n and $n-1$, where the additional factor c can be chosen freely. The proof of Lemma 10 is by induction on the degree of q and its structure resembles the structure of the following algorithm to compute r. Using the Binomial Theorem, one can verify that $q - s + c \cdot s[n/n-1]$ has a smaller degree than q, which is crucial for the proof of Lemma 10 and termination of Algorithm 1.

Algorithm 1. compute_r

Input: $q = \sum_{i=0}^{d} c_i \cdot n^i \in \mathbb{Q}[n]$, $c \in \mathbb{Q}$
Result: $r \in \mathbb{Q}[n]$ such that $q = r - c \cdot r[n/n-1]$
if $d = 0$ **then**
  if $c = 1$ **then return** $c_0 \cdot n$ **else return** $\frac{c_0}{1-c}$
else
  if $c = 1$ **then** $s \leftarrow \frac{c_d \cdot n^{d+1}}{d+1}$ **else** $s \leftarrow \frac{c_d \cdot n^d}{1-c}$
  return $s +$ compute_$r(\, q - s + c \cdot s[n/n-1],\ c\,)$

Example 11. *As an example, consider $q = 1$ (i.e., $c_0 = 1$) and $c = 4$. Then we search for an r such that $q = r - c \cdot r[n/n-1]$, i.e., $1 = r - 4 \cdot r[n/n-1]$. According to Algorithm 1, the solution is $r = \frac{c_0}{1-c} = -\frac{1}{3}$.*

Lemma 12 (Closure of $\mathbb{PE}$ under Sums of Products and Exponentials). *If $m \in \mathbb{N}$ and $p \in \mathbb{PE}[\overline{x}]$, then one can compute a $q \in \mathbb{PE}[\overline{x}]$ which is equivalent to $\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1]$.*

Proof. Let $p = \sum_{j=1}^{\ell} [\![\psi_j]\!] \cdot \alpha_j \cdot n^{a_j} \cdot b_j^n$. We have:

$$\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1] = \sum_{j=1}^{\ell} \sum_{i=1}^{n} [\![\psi_j]\!]\,(i-1) \cdot m^{n-i} \cdot \alpha_j \cdot (i-1)^{a_j} \cdot b_j^{i-1} \tag{3}$$

As $\mathbb{PE}[\overline{x}]$ is closed under addition, it suffices to show that we can compute an equivalent poly-exponential expression for any expression of the form

$$\textstyle\sum_{i=1}^{n} [\![\psi]\!]\,(i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1}. \tag{4}$$

We first regard the case $m = 0$. Here, the expression (4) can be simplified to

$$[\![n \neq 0]\!] \cdot [\![\psi[n/n-1]]\!] \cdot \alpha \cdot (n-1)^a \cdot b^{n-1}. \tag{5}$$

Clearly, there is a $\psi' \in \mathcal{C}$ such that $[\![\psi']\!]$ is equivalent to $[\![n \neq 0]\!] \cdot [\![\psi[n/n-1]]\!]$. Moreover, $\alpha \cdot b^{n-1} = \frac{\alpha}{b} \cdot b^n$ where $\frac{\alpha}{b} \in \mathbb{Af}[\overline{x}]$. Hence, due to the Binomial Theorem

$$[\![n \neq 0]\!] \cdot [\![\psi[n/n-1]]\!] \cdot \alpha \cdot (n-1)^a \cdot b^{n-1} = \textstyle\sum_{i=0}^{a} [\![\psi']\!] \cdot \frac{\alpha}{b} \cdot \binom{a}{i} \cdot (-1)^i \cdot n^{a-i} \cdot b^n \tag{6}$$

which is a poly-exponential expression as $\frac{\alpha}{b} \cdot \binom{a}{i} \cdot (-1)^i \in \mathbb{Af}[\overline{x}]$.

From now on, let $m \geq 1$. If ψ contains a positive literal $n = c$, then we get

$$\left.\begin{array}{ll} & \sum_{i=1}^{n} [\![\psi]\!]\,(i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \\ = & \sum_{i=1}^{n} [\![n > i-1]\!] \cdot [\![\psi]\!]\,(i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \quad (\dagger) \\ = & [\![n > c]\!] \cdot [\![\psi]\!]\,(c) \cdot m^{n-c-1} \cdot \alpha \cdot c^a \cdot b^c \quad (\dagger\dagger) \\ = & \begin{cases} 0, & \text{if } [\![\psi]\!]\,(c) = 0 \\ [\![n > c]\!] \cdot \frac{1}{m^{c+1}} \cdot \alpha \cdot c^a \cdot b^c \cdot m^n, & \text{if } [\![\psi]\!]\,(c) = 1 \end{cases} \\ \in & \mathbb{PE}[\overline{x}] \quad (\text{since } \tfrac{1}{m^{c+1}} \cdot \alpha \cdot c^a \cdot b^c \in \mathbb{Af}[\overline{x}]). \end{array}\right\} \tag{7}$$

The step marked with (†) holds as we have $[\![n > i-1]\!] = 1$ for all $i \in \{1, \dots, n\}$ and the step marked with (††) holds since $i \neq c+1$ implies $[\![\psi]\!](i-1) = 0$. If ψ does not contain a positive literal, then let c be the maximal constant that occurs in ψ or -1 if ψ is empty. We get:

$$\left.\begin{array}{l} \sum_{i=1}^{n} [\![\psi]\!](i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \\ = \sum_{i=1}^{n} [\![n > i-1]\!] \cdot [\![\psi]\!](i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \quad (\dagger) \\ = \sum_{i=1}^{c+1} [\![n > i-1]\!] \cdot [\![\psi]\!](i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \\ \quad + \sum_{i=c+2}^{n} m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \end{array}\right\} \tag{8}$$

Again, the step marked with (†) holds since we have $[\![n > i-1]\!] = 1$ for all $i \in \{1, \dots, n\}$. The last step holds as $i \geq c+2$ implies $[\![\psi]\!](i-1) = 1$. Similar to the case where ψ contains a positive literal, we can compute a poly-exponential expression which is equivalent to the first addend. We have

$$\begin{array}{l} \sum_{i=1}^{c+1} [\![n > i-1]\!] \cdot [\![\psi]\!](i-1) \cdot m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \\ = \sum\limits_{\substack{1 \leq i \leq c+1 \\ [\![\psi]\!](i-1)=1}} [\![n > i-1]\!] \cdot \frac{1}{m^i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \cdot m^n \end{array} \tag{9}$$

which is a poly-exponential expression as $\frac{1}{m^i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \in \mathbb{A}\mathbb{f}[\overline{x}]$. For the second addend, we have:

$$\left.\begin{array}{l} \sum_{i=c+2}^{n} m^{n-i} \cdot \alpha \cdot (i-1)^a \cdot b^{i-1} \\ = \frac{\alpha}{b} \cdot m^n \cdot \sum_{i=c+2}^{n} (i-1)^a \cdot \left(\frac{b}{m}\right)^i \\ = \frac{\alpha}{b} \cdot m^n \cdot \sum_{i=c+2}^{n} (r[n/i] - \frac{m}{b} \cdot r[n/i-1]) \cdot \left(\frac{b}{m}\right)^i \text{ (Lemma 10 with } c = \frac{m}{b}) \\ = \frac{\alpha}{b} \cdot m^n \cdot \left(\sum_{i=c+2}^{n} r[n/i] \cdot \left(\frac{b}{m}\right)^i - \sum_{i=c+2}^{n} \frac{m}{b} \cdot r[n/i-1] \cdot \left(\frac{b}{m}\right)^i\right) \\ = \frac{\alpha}{b} \cdot m^n \cdot \left(\sum_{i=c+2}^{n} r[n/i] \cdot \left(\frac{b}{m}\right)^i - \sum_{i=c+1}^{n-1} r[n/i] \cdot \left(\frac{b}{m}\right)^i\right) \\ = \frac{\alpha}{b} \cdot m^n \cdot [\![n > c+1]\!] \cdot (r \cdot \left(\frac{b}{m}\right)^n - r[n/c+1] \cdot \left(\frac{b}{m}\right)^{c+1}) \\ = [\![n > c+1]\!] \cdot \frac{\alpha}{b} \cdot r \cdot b^n - [\![n > c+1]\!] \cdot r[n/c+1] \cdot \left(\frac{b}{m}\right)^{c+1} \cdot \frac{\alpha}{b} \cdot m^n \end{array}\right\} \tag{10}$$

Lemma 10 ensures $r \in \mathbb{Q}[n]$, i.e., we have $r = \sum_{i=0}^{d_r} m_i \cdot n^i$ for some $d_r \in \mathbb{N}$ and $m_i \in \mathbb{Q}$. Thus, $r[n/c+1] \cdot \left(\frac{b}{m}\right)^{c+1} \cdot \frac{\alpha}{b} \in \mathbb{A}\mathbb{f}[\overline{x}]$ which implies $[\![n > c+1]\!] \cdot r[n/c+1] \cdot \left(\frac{b}{m}\right)^{c+1} \cdot \frac{\alpha}{b} \cdot m^n \in \mathbb{PE}[\overline{x}]$. It remains to show that the addend $[\![n > c+1]\!] \cdot \frac{\alpha}{b} \cdot r \cdot b^n$ is equivalent to a poly-exponential expression. As $\frac{\alpha}{b} \cdot m_i \in \mathbb{A}\mathbb{f}[\overline{x}]$, we have

$$[\![n > c+1]\!] \cdot \frac{\alpha}{b} \cdot r \cdot b^n = \sum_{i=0}^{d_r} [\![n > c+1]\!] \cdot \frac{\alpha}{b} \cdot m_i \cdot n^i \cdot b^n \in \mathbb{PE}[\overline{x}]. \tag{11}$$

□

The proof of Lemma 12 gives rise to a corresponding algorithm.

Algorithm 2. symbolic_sum

Input: $m \in \mathbb{N}$, $p \in \mathbb{PE}[\overline{x}]$
Result: $q \in \mathbb{PE}[\overline{x}]$ which is equivalent to $\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1]$
rearrange $\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1]$ to $\sum_{j=1}^{\ell} p_j$ as in (3)
foreach $p_j \in \{p_1, \ldots, p_\ell\}$ **do**
  if $m = 0$ **then** compute q_j as in (5) and (6)
  else if $p_j = [\![\ldots \wedge n = c \wedge \ldots]\!] \cdot \ldots$ **then** compute q_j as in (7)
  else
    - split p_j into two sums $p_{j,1}$ and $p_{j,2}$ as in (8)
    - compute $q_{j,1}$ from $p_{j,1}$ as in (9)
    - compute $q_{j,2}$ from $p_{j,2}$ as in (10) and (11) using Algorithm 1
    - $q_j \leftarrow q_{j,1} + q_{j,2}$
return $\sum_{j=1}^{\ell} q_j$

Example 13. *We compute an equivalent poly-exponential expression for*

$$\sum_{i=1}^{n} 4^{n-i} \cdot ([\![n = 0]\!] \cdot 2 \cdot w \ + \ [\![n \neq 0]\!] \cdot 4 \ - \ 2)\,[n/i-1] \tag{12}$$

where w is a variable. (It will later on be needed to compute a closed form for Example 4, see Example 18.) According to Algorithm 2 and (3), we get

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} 4^{n-i} \cdot ([\![n = 0]\!] \cdot 2 \cdot w \ + \ [\![n \neq 0]\!] \cdot 4 \ - \ 2)\,[n/i-1] \\ &= \textstyle\sum_{i=1}^{n} 4^{n-i} \cdot ([\![i-1 = 0]\!] \cdot 2 \cdot w \ + \ [\![i-1 \neq 0]\!] \cdot 4 \ - \ 2) \\ &= p_1 + p_2 + p_3 \end{aligned}$$

with $p_1 = \sum_{i=1}^{n} [\![i-1=0]\!] \cdot 4^{n-i} \cdot 2 \cdot w$, $p_2 = \sum_{i=1}^{n} [\![i-1 \neq 0]\!] \cdot 4^{n-i} \cdot 4$, and $p_3 = \sum_{i=1}^{n} 4^{n-i} \cdot (-2)$. We search for $q_1, q_2, q_3 \in \mathbb{PE}[w]$ that are equivalent to p_1, p_2, p_3, i.e., $q_1 + q_2 + q_3$ is equivalent to (12). We only show how to compute q_2 (and omit the computation of $q_1 = [\![n \neq 0]\!] \cdot \frac{1}{2} \cdot w \cdot 4^n$ and $q_3 = \frac{2}{3} - \frac{2}{3} \cdot 4^n$). Analogously to (8), we get:

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} [\![i-1 \neq 0]\!] \cdot 4^{n-i} \cdot 4 \\ &= \textstyle\sum_{i=1}^{n} [\![n > i-1]\!] \cdot [\![i-1 \neq 0]\!] \cdot 4^{n-i} \cdot 4 \\ &= \textstyle\sum_{i=1}^{1} [\![n > i-1]\!] \cdot [\![i-1 \neq 0]\!] \cdot 4^{n-1} \cdot 4 \quad + \quad \sum_{i=2}^{n} 4^{n-i} \cdot 4 \end{aligned}$$

The next step is to rearrange the first sum as in (9). In our example, it directly simplifies to 0 and hence we obtain

$$\textstyle\sum_{i=1}^{1} [\![n > i-1]\!] \cdot [\![i-1 \neq 0]\!] \cdot 4^{n-1} \cdot 4 + \sum_{i=2}^{n} 4^{n-i} \cdot 4 = \sum_{i=2}^{n} 4^{n-i} \cdot 4.$$

Finally, by applying the steps from (10) we get:

$$\begin{aligned}
&\textstyle\sum_{i=2}^{n} 4^{n-i} \cdot 4 \\
&= 4 \cdot 4^n \cdot \textstyle\sum_{i=2}^{n} \left(\frac{1}{4}\right)^i \\
&= 4 \cdot 4^n \cdot \textstyle\sum_{i=2}^{n} \left(-\frac{1}{3} - 4 \cdot \left(-\frac{1}{3}\right)\right) \cdot \left(\frac{1}{4}\right)^i && (\dagger) \\
&= 4 \cdot 4^n \cdot \left(\textstyle\sum_{i=2}^{n} \left(-\frac{1}{3}\right) \cdot \left(\frac{1}{4}\right)^i - \sum_{i=2}^{n} 4 \cdot \left(-\frac{1}{3}\right) \cdot \left(\frac{1}{4}\right)^i\right) \\
&= 4 \cdot 4^n \cdot \left(\textstyle\sum_{i=2}^{n} \left(-\frac{1}{3}\right) \cdot \left(\frac{1}{4}\right)^i - \sum_{i=1}^{n-1} \left(-\frac{1}{3}\right) \cdot \left(\frac{1}{4}\right)^i\right) \\
&= 4 \cdot 4^n \cdot [\![n > 1]\!] \cdot \left(\left(-\tfrac{1}{3}\right) \cdot \left(\tfrac{1}{4}\right)^n - \left(-\tfrac{1}{3}\right) \cdot \tfrac{1}{4}\right) \\
&= [\![n > 1]\!] \cdot \left(-\tfrac{4}{3}\right) + [\![n > 1]\!] \cdot \tfrac{1}{3} \cdot 4^n \\
&= q_2
\end{aligned}$$

The step marked with (†) holds by Lemma 10 with $q = 1$ and $c = 4$. Thus, we have $r = -\frac{1}{3}$, cf. Example 11.

Recall that our goal is to compute closed forms for loops. As a first step, instead of the n-fold update function $h(n, \overline{x}) = f^n(\overline{x})$ of (1) where f is the update of (1), we consider a recursive update function for a single variable $x \in \overline{x}$:

$$g(0, \overline{x}) = x \quad \text{and} \quad g(n, \overline{x}) = m \cdot g(n-1, \overline{x}) + p[n/n-1] \quad \text{for all } n > 0$$

Here, $m \in \mathbb{N}$ and $p \in \mathbb{PE}[\overline{x}]$. Using Lemma 12, it is easy to show that g can be represented by a poly-exponential expression.

Lemma 14 (Closed Form for Single Variables). *If $x \in \overline{x}$, $m \in \mathbb{N}$, and $p \in \mathbb{PE}[\overline{x}]$, then one can compute a $q \in \mathbb{PE}[\overline{x}]$ which satisfies*

$$q\,[n/0] = x \quad \textit{and} \quad q = (m \cdot q + p)\,[n/n-1] \quad \textit{for all } n > 0.$$

Proof. It suffices to find a $q \in \mathbb{PE}[\overline{x}]$ that satisfies

$$q = m^n \cdot x + \textstyle\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1]. \tag{13}$$

To see why (13) is sufficient, note that (13) implies

$$q[n/0] \quad = \quad m^0 \cdot x + \textstyle\sum_{i=1}^{0} m^{0-i} \cdot p[n/i-1] \quad = \quad x$$

and for $n > 0$, (13) implies

$$\begin{aligned}
q &= m^n \cdot x + \textstyle\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1] \\
&= m^n \cdot x + \left(\textstyle\sum_{i=1}^{n-1} m^{n-i} \cdot p[n/i-1]\right) + p[n/n-1] \\
&= m \cdot \left(m^{n-1} \cdot x + \textstyle\sum_{i=1}^{n-1} m^{n-i-1} \cdot p[n/i-1]\right) + p[n/n-1] \\
&= m \cdot q[n/n-1] + p[n/n-1] \\
&= (m \cdot q + p)[n/n-1].
\end{aligned}$$

By Lemma 12, we can compute a $q' \in \mathbb{PE}[\overline{x}]$ such that

$$m^n \cdot x + \textstyle\sum_{i=1}^{n} m^{n-i} \cdot p[n/i-1] \quad = \quad m^n \cdot x + q'.$$

Moreover,

$$\text{if } m = 0, \text{ then } m^n \cdot x = [\![n = 0]\!] \cdot x \in \mathbb{PE}[\overline{x}] \text{ and} \tag{14}$$

$$\text{if } m > 0, \text{ then } m^n \cdot x \in \mathbb{PE}[\overline{x}]. \tag{15}$$

So both addends are equivalent to poly-exponential expressions. □

Example 15. *We show how to compute the closed forms for the variables w and x from Example 4. We first consider the assignment $w \leftarrow 2$, i.e., we want to compute a $q_w \in \mathbb{PE}[w, x, y, z]$ with $q_w[n/0] = w$ and $q_w = (m_w \cdot q_w + p_w)\,[n/n-1]$ for $n > 0$, where $m_w = 0$ and $p_w = 2$. According to (13) and (14), q_w is*

$$m_w^n \cdot w + \textstyle\sum_{i=1}^{n} m_w^{n-i} \cdot p_w[n/i-1] = 0^n \cdot w + \sum_{i=1}^{n} 0^{n-i} \cdot 2 = [\![n = 0]\!] \cdot w + [\![n \neq 0]\!] \cdot 2.$$

For the assignment $x \leftarrow x + 2$, we search for a q_x such that $q_x[n/0] = x$ and $q_x = (m_x \cdot q_x + p_x)\,[n/n-1]$ for $n > 0$, where $m_x = 1$ and $p_x = 2$. By (13), q_x is

$$m_x^n \cdot x + \textstyle\sum_{i=1}^{n} m_x^{n-i} \cdot p_x[n/i-1] = 1^n \cdot x + \sum_{i=1}^{n} 1^{n-i} \cdot 2 = x + 2 \cdot n.$$

The restriction to triangular matrices now allows us to generalize Lemma 14 to vectors of variables. The reason is that due to triangularity, the update of each program variable x_i only depends on the previous values of $x_1, \ldots, x_i$. So when regarding x_i, we can assume that we already know the closed forms for $x_1, \ldots, x_{i-1}$. This allows us to find closed forms for one variable after the other by applying Lemma 14 repeatedly. In other words, it allows us to find a vector $\overline{q}$ of poly-exponential expressions that satisfies

$$\overline{q}\,[n/0] = \overline{x} \quad \text{and} \quad \overline{q} = A\,\overline{q}[n/n-1] + \overline{a} \quad \text{for all } n > 0.$$

To prove this claim, we show the more general Lemma 16. For all $i_1, \ldots, i_k \in \{1, \ldots, m\}$, we define $[z_1, \ldots, z_m]_{i_1,\ldots,i_k} = [z_{i_1}, \ldots, z_{i_k}]$ (and the notation $\overline{y}_{i_1,\ldots,i_k}$ for column vectors is defined analogously). Moreover, for a matrix A, A_i is A's i^{th} row and $A_{i_1,\ldots,i_n;j_1,\ldots,j_k}$ is the matrix with rows $(A_{i_1})_{j_1,\ldots,j_k}, \ldots, (A_{i_n})_{j_1,\ldots,j_k}$. So for $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$, we have $A_{1,2;1,3} = \begin{bmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{bmatrix}$.

Lemma 16. (Closed Forms for Vectors of Variables). *If $\overline{x}$ is a vector of at least $d \geq 1$ pairwise different variables, $A \in \mathbb{Z}^{d \times d}$ is triangular with $A_{i;i} \geq 0$ for all $1 \leq i \leq d$, and $\overline{p} \in \mathbb{PE}[\overline{x}]^d$, then one can compute $\overline{q} \in \mathbb{PE}[\overline{x}]^d$ such that:*

$$\overline{q}\,[n/0] = \overline{x}_{1,\ldots,d} \quad \textit{and} \tag{16}$$

$$\overline{q} = (A\,\overline{q} + \overline{p})\,[n/n-1] \quad \textit{for all } n > 0 \tag{17}$$

Proof. Assume that A is lower triangular (the case that A is upper triangular works analogously). We use induction on d. For any $d \geq 1$ we have:

$$\begin{array}{rll}
& \overline{q} = (A\,\overline{q} + \overline{p})\ [n/n-1] & \\
\iff & \overline{q}_j = (A_j \cdot \overline{q} + \overline{p}_j)\ [n/n-1] & \text{for all } 1 \leq j \leq d \\
\iff & \overline{q}_j = (A_{j;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + A_{j;1} \cdot \overline{q}_1 + \overline{p}_j)\ [n/n-1] & \text{for all } 1 \leq j \leq d \\
\iff & \overline{q}_1 = (A_{1;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + A_{1;1} \cdot \overline{q}_1 + \overline{p}_1)\ [n/n-1] \wedge & \\
& \overline{q}_j = (A_{j;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + A_{j;1} \cdot \overline{q}_1 + \overline{p}_j)\ [n/n-1] & \text{for all } 1 < j \leq d \\
\iff & \overline{q}_1 = (A_{1;1} \cdot \overline{q}_1 + \overline{p}_1)\ [n/n-1] & \wedge \\
& \overline{q}_j = (A_{j;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + A_{j;1} \cdot \overline{q}_1 + \overline{p}_j)\ [n/n-1] & \text{for all } 1 < j \leq d
\end{array}$$

The last step holds as A is lower triangular. By Lemma 14, we can compute a $\overline{q}_1 \in \mathbb{PE}[\overline{x}]$ that satisfies

$$\overline{q}_1[n/0] = \overline{x}_1 \quad \text{and} \quad \overline{q}_1 = (A_{1;1} \cdot \overline{q}_1 + \overline{p}_1)\ [n/n-1] \quad \text{for all } n > 0.$$

In the induction base ($d = 1$), there is no j with $1 < j \leq d$. In the induction step ($d > 1$), it remains to show that we can compute $\overline{q}_{2,\ldots,d}$ such that

$$\overline{q}_j[n/0] = \overline{x}_j \quad \text{and} \quad \overline{q}_j = (A_{j;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + A_{j;1} \cdot \overline{q}_1 + \overline{p}_j)\ [n/n-1]$$

for all $n > 0$ and all $1 < j \leq d$, which is equivalent to

$$\begin{array}{l}
\overline{q}_{2,\ldots,d}[n/0] = \overline{x}_{2,\ldots,d} \quad \text{and} \\
\qquad \overline{q}_{2,\ldots,d} = (A_{2,\ldots,d;2,\ldots,d} \cdot \overline{q}_{2,\ldots,d} + \begin{bmatrix} A_{2;1} \\ \vdots \\ A_{d;1} \end{bmatrix} \cdot \overline{q}_1 + \overline{p}_{2,\ldots,d})\ [n/n-1]
\end{array}$$

for all $n > 0$. As $A_{j;1} \cdot \overline{q}_1 + \overline{p}_j \in \mathbb{PE}[\overline{x}]$ for each $2 \leq j \leq d$, the claim follows from the induction hypothesis. □

Together, Lemmas 14 and 16 and their proofs give rise to the following algorithm to compute a solution for (16) and (17). It computes a closed form $\overline{q}_1$ for $\overline{x}_1$ as in the proof of Lemma 14, constructs the argument $\overline{p}$ for the recursive call based on A, $\overline{q}_1$, and the current value of $\overline{p}$ as in the proof of Lemma 16, and then determines the closed form for $\overline{x}_{2,\ldots,d}$ recursively.

Algorithm 3. closed_form

Input: $\overline{x}_{1,\ldots,d}$, $A \in \mathbb{Z}^{d \times d}$ where $A_{i;i} \geq 0$ for all $1 \leq i \leq d$, $\overline{p} \in \mathbb{PE}[\overline{x}]^d$
Result: $\overline{q} \in \mathbb{PE}[\overline{x}]^d$ which satisfies (16) & (17) for the given $\overline{x}$, A, and $\overline{p}$
$q \leftarrow$ symbolic_sum$(A_{1;1}, \overline{p}_1)$ (cf. Algorithm 2)
if $A_{1;1} = 0$ **then** $\overline{q}_1 \leftarrow [\![n = 0]\!] \cdot \overline{x}_1 + q$ **else** $\overline{q}_1 \leftarrow A_{1;1}^n \cdot \overline{x}_1 + q$ (cf. (13–15))
if $d > 1$ **then**
$\overline{q}_{2,\ldots,d} \leftarrow$ closed_form$(\overline{x}_{2,\ldots,d}, A_{2,\ldots,d;2,\ldots,d}, \begin{bmatrix} A_{2;1} \\ \vdots \\ A_{d;1} \end{bmatrix} \cdot \overline{q}_1 + \overline{p}_{2,\ldots,d})$
return $\overline{q}$

We can now prove the main theorem of this section.

Theorem 17 (Closed Forms for nnt-Loops). *One can compute a closed form for every nnt-loop. In other words, if $f : \mathbb{Z}^d \to \mathbb{Z}^d$ is the update function of an nnt-loop with the variables $\overline{x}$, then one can compute a $\overline{q} \in \mathbb{PE}[\overline{x}]^d$ such that $\overline{q}[n/c] = f^c(\overline{x})$ for all $c \in \mathbb{N}$.*

Proof. Consider an nnt-loop of the form (1). By Lemma 16, we can compute a $\overline{q} \subseteq \mathbb{PE}[\overline{x}]^d$ that satisfies

$$\overline{q}[n/0] = \overline{x} \quad \text{and} \quad \overline{q} = (A\,\overline{q} + \overline{a})\;[n/n-1] \quad \text{for all } n > 0.$$

We prove $f^c(\overline{x}) = \overline{q}[n/c]$ by induction on $c \in \mathbb{N}$. If $c = 0$, we get

$$f^c(\overline{x}) = f^0(\overline{x}) = \overline{x} = \overline{q}[n/0] = \overline{q}[n/c].$$

If $c > 0$, we get:

$$\begin{aligned} f^c(\overline{x}) &= A\,f^{c-1}(\overline{x}) + \overline{a} && \text{by definition of } f \\ &= A\,\overline{q}[n/c-1] + \overline{a} && \text{by the induction hypothesis} \\ &= (A\,\overline{q} + \overline{a})\;[n/c-1] && \text{as } \overline{a} \in \mathbb{Z}^d \text{ does not contain } n \\ &= \overline{q}[n/c] \end{aligned}$$

□

So invoking Algorithm 3 on $\overline{x}$, A, and $\overline{a}$ yields the closed form of an nnt-loop (1).

Example 18. *We show how to compute the closed form for Example 4. For*

$$y \leftarrow 2 \cdot w + 4 \cdot y - 2,$$

we obtain

$$\begin{aligned} q_y &= (4 \cdot q_y + 2 \cdot q_w - 2)\;[n/n-1] \\ &= 4^n \cdot y + \textstyle\sum_{i=1}^{n} 4^{n-i} \cdot (2 \cdot q_w - 2)\;[n/i-1] && (\textit{by } (13)) \\ &= y \cdot 4^n + \textstyle\sum_{i=1}^{n} 4^{n-i} \cdot ([\![n = 0]\!] \cdot 2 \cdot w + [\![n \neq 0]\!] \cdot 4 - 2)\;[n/i-1] && (\textit{see Example 15}) \\ &= q_0 + q_1 + q_2 + q_3 && (\textit{see Example 13}) \end{aligned}$$

where $q_0 = y \cdot 4^n$. For $z \leftarrow x + 1$, we get

$$\begin{aligned} q_z &= (q_x + 1)\;[n/n-1] \\ &= 0^n \cdot z + \textstyle\sum_{i=1}^{n} 0^{n-i} \cdot (q_x + 1)\;[n/i-1] && (\textit{by } (13)) \\ &= [\![n = 0]\!] \cdot z + [\![n \neq 0]\!] \cdot (q_x[n/n-1] + 1) \\ &= [\![n = 0]\!] \cdot z + [\![n \neq 0]\!] \cdot ((x + 2 \cdot n)\;[n/n-1] + 1) && (\textit{see Example 15}) \\ &= [\![n = 0]\!] \cdot z + [\![n \neq 0]\!] \cdot (x - 1) + [\![n \neq 0]\!] \cdot 2 \cdot n. \end{aligned}$$

So the closed form of Example 4 for the values of the variables after n iterations is:

$$\begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} [\![n = 0]\!] \cdot w + [\![n \neq 0]\!] \cdot 2 \\ x + 2 \cdot n \\ q_0 + q_1 + q_2 + q_3 \\ [\![n = 0]\!] \cdot z + [\![n \neq 0]\!] \cdot (x - 1) + [\![n \neq 0]\!] \cdot 2 \cdot n \end{bmatrix}$$

4 Deciding Non-Termination of nnt-Loops

Our proof uses the notion of *eventual non-termination* [4,14]. Here, the idea is to disregard the condition of the loop during a finite prefix of the program run.

Definition 19 (Eventual Non-Termination). *A vector* $\overline{c} \in \mathbb{Z}^d$ witnesses eventual non-termination *of (1) if*

$$\exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \varphi[\overline{x}/f^n(\overline{c})].$$

If there is such a witness, then (1) is eventually non-terminating.

Clearly, (1) is non-terminating iff (1) is eventually non-terminating [14]. Now Theorem 17 gives rise to an alternative characterization of eventual non-termination in terms of the closed form $\overline{q}$ instead of $f^n(\overline{c})$.

Corollary 20 (Expressing Non-Termination with $\mathbb{PE}$). *If $\overline{q}$ is the closed form of (1), then $\overline{c} \in \mathbb{Z}^d$ witnesses eventual non-termination iff*

$$\exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \varphi[\overline{x}/\overline{q}][\overline{x}/\overline{c}]. \tag{18}$$

Proof. Immediate, as $\overline{q}$ is equivalent to $f^n(\overline{x})$. □

So to prove that termination of nnt-loops is decidable, we will use Corollary 20 to show that the existence of a witness for eventual non-termination is decidable. To do so, we first eliminate the factors $[\![\psi]\!]$ from the closed form $\overline{q}$. Assume that $\overline{q}$ has at least one factor $[\![\psi]\!]$ where ψ is non-empty (otherwise, all factors $[\![\psi]\!]$ are equivalent to 1) and let c be the maximal constant that occurs in such a factor. Then all addends $[\![\psi]\!] \cdot \alpha \cdot n^a \cdot b^n$ where ψ contains a positive literal become 0 and all other addends become $\alpha \cdot n^a \cdot b^n$ if $n > c$. Thus, as we can assume $n_0 > c$ in (18) without loss of generality, all factors $[\![\psi]\!]$ can be eliminated when checking eventual non-termination.

Corollary 21 Removing $[\![\psi]\!]$ from $\mathbb{PE}$s). *Let $\overline{q}$ be the closed form of an nnt-loop (1). Let $\overline{q}_{norm}$ result from $\overline{q}$ by removing all addends $[\![\psi]\!] \cdot \alpha \cdot n^a \cdot b^n$ where ψ contains a positive literal and by replacing all addends $[\![\psi]\!] \cdot \alpha \cdot n^a \cdot b^n$ where ψ does not contain a positive literal by $\alpha \cdot n^a \cdot b^n$. Then $\overline{c} \in \mathbb{Z}^d$ is a witness for eventual non-termination iff*

$$\exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \varphi[\overline{x}/\overline{q}_{norm}][\overline{x}/\overline{c}]. \tag{19}$$

By removing the factors $[\![\psi]\!]$ from the closed form $\overline{q}$ of an nnt-loop, we obtain *normalized* poly-exponential expressions.

Definition 22 (Normalized $\mathbb{PE}$s). *We call $p \in \mathbb{PE}[\overline{x}]$* normalized *if it is in*

$$\mathbb{NPE}[\overline{x}] = \left\{ \textstyle\sum_{j=1}^{\ell} \alpha_j \cdot n^{a_j} \cdot b_j^n \;\middle|\; \ell, a_j \in \mathbb{N},\ \alpha_j \in \mathbb{Af}[\overline{x}],\ b_j \in \mathbb{N}_{\geq 1} \right\}.$$

W.l.o.g., we always assume $(b_i, a_i) \neq (b_j, a_j)$ for all $i, j \in \{1, \ldots, \ell\}$ with $i \neq j$. We define $\mathbb{NPE} = \mathbb{NPE}[\varnothing]$, i.e., we have $p \in \mathbb{NPE}$ if $\alpha_j \in \mathbb{Q}$ for all $1 \leq j \leq \ell$.

Example 23. *We continue Example 18. By omitting the factors* $[\![\psi]\!]$,

$$\begin{array}{ll} q_w = [\![n=0]\!]\cdot w + [\![n\neq 0]\!]\cdot 2 & \text{becomes } 2,\\ q_z = [\![n=0]\!]\cdot z + [\![n\neq 0]\!]\cdot (x-1) + [\![n\neq 0]\!]\cdot 2\cdot n & \text{becomes } x-1+2\cdot n, \end{array}$$

and $q_x = x + 2\cdot n, q_0 = y\cdot 4^n$, *and* $q_3 = \frac{2}{3} - \frac{2}{3}\cdot 4^n$ *remain unchanged. Moreover,*

$$\begin{array}{lll} q_1 = [\![n\neq 0]\!]\cdot \frac{1}{2}\cdot w\cdot 4^n & \text{becomes } \frac{1}{2}\cdot w\cdot 4^n & \text{and}\\ q_2 = [\![n>1]\!]\cdot \left(-\frac{4}{3}\right) + [\![n>1]\!]\cdot \frac{1}{3}\cdot 4^n & \text{becomes } \left(-\frac{4}{3}\right) + \frac{1}{3}\cdot 4^n. & \end{array}$$

Thus, $q_y = q_0 + q_1 + q_2 + q_3$ *becomes*

$$y\cdot 4^n + \tfrac{1}{2}\cdot w\cdot 4^n - \tfrac{4}{3} + \tfrac{1}{3}\cdot 4^n + \tfrac{2}{3} - \tfrac{2}{3}\cdot 4^n = 4^n\cdot\left(y - \tfrac{1}{3} + \tfrac{1}{2}\cdot w\right) - \tfrac{2}{3}.$$

Let $\sigma = \left[w/2,\ x/x + 2\cdot n,\ y/4^n\cdot\left(y - \frac{1}{3} + \frac{1}{2}\cdot w\right) - \frac{2}{3},\ z/x - 1 + 2\cdot n\right]$. *Then we get that Example 2 is non-terminating iff there are* $w, x, y, z \in \mathbb{Z}, n_0 \in \mathbb{N}$ *such that*

$$\begin{array}{ll} (y+z)\,\sigma > 0 \wedge (-w - 2\cdot y + x)\,\sigma > 0 & \iff\\ 4^n\cdot\left(y - \frac{1}{3} + \frac{1}{2}\cdot w\right) - \frac{2}{3} + x - 1 + 2\cdot n > 0 \quad \wedge & \\ \quad -2 - 2\cdot\left(4^n\cdot\left(y - \frac{1}{3} + \frac{1}{2}\cdot w\right) - \frac{2}{3}\right) + x + 2\cdot n > 0 & \iff\\ p_1^{\varphi} > 0 \wedge p_2^{\varphi} > 0 & \end{array}$$

holds for all $n > n_0$ *where*

$$\begin{array}{l} p_1^{\varphi} = 4^n\cdot\left(y - \frac{1}{3} + \frac{1}{2}\cdot w\right) + 2\cdot n + x - \frac{5}{3} \text{ and}\\ p_2^{\varphi} = 4^n\cdot\left(\frac{2}{3} - 2\cdot y - w\right) + 2\cdot n + x - \frac{2}{3}. \end{array}$$

Recall that the loop condition φ is a conjunction of inequalities of the form $\alpha > 0$ where $\alpha \in \mathbb{A}\mathrm{f}[\overline{x}]$. Thus, $\varphi[\overline{x}/\overline{q}_{norm}]$ is a conjunction of inequalities $p > 0$ where $p \in \mathbb{NPE}[\overline{x}]$ and we need to decide if there is an instantiation of these inequalities that is valid "for large enough n". To do so, we order the coefficients α_j of the addends $\alpha_j \cdot n^{a_j} \cdot b_j^n$ of normalized poly-exponential expressions according to the addend's asymptotic growth when increasing n. Lemma 24 shows that $\alpha_2 \cdot n^{a_2} \cdot b_2^n$ grows faster than $\alpha_1 \cdot n^{a_1} \cdot b_1^n$ iff $b_2 > b_1$ or both $b_2 = b_1$ and $a_2 > a_1$.

Lemma 24 (Asymptotic Growth). *Let* $b_1, b_2 \in \mathbb{N}_{\geq 1}$ *and* $a_1, a_2 \in \mathbb{N}$. *If* $(b_2, a_2) >_{lex} (b_1, a_1)$, *then* $\mathcal{O}(n^{a_1}\cdot b_1^n) \subsetneq \mathcal{O}(n^{a_2}\cdot b_2^n)$. *Here,* $>_{lex}$ *is the lexicographic order, i.e.,* $(b_2, a_2) >_{lex} (b_1, a_1)$ *iff* $b_2 > b_1$ *or* $b_2 = b_1 \wedge a_2 > a_1$.

Proof. By considering the cases $b_2 > b_1$ and $b_2 = b_1$ separately, the claim can easily be deduced from the definition of $\mathcal{O}$. □

Definition 25 (Ordering Coefficients). Marked coefficients *are of the form* $\alpha^{(b,a)}$ *where* $\alpha \in \mathbb{A}\mathrm{f}[\overline{x}], b \in \mathbb{N}_{\geq 1}$, *and* $a \in \mathbb{N}$. *We define* $\mathrm{unmark}(\alpha^{(b,a)}) = \alpha$ *and* $\alpha_2^{(b_2,a_2)} \succ \alpha_1^{(b_1,a_1)}$ *if* $(b_2, a_2) >_{lex} (b_1, a_1)$. *Let*

$$p = \textstyle\sum_{j=1}^{\ell} \alpha_j \cdot n^{a_j} \cdot b_j^n \in \mathbb{NPE}[\overline{x}],$$

where $\alpha_j \neq 0$ for all $1 \leq j \leq \ell$. The marked coefficients of p are

$$\text{coeffs}(p) = \begin{cases} \left\{0^{(1,0)}\right\}, & \text{if } \ell = 0 \\ \left\{\alpha_j^{(b_j,a_j)} \;\middle|\; 0 \leq j \leq \ell\right\}, & \text{otherwise.} \end{cases}$$

Example 26. *In Example 23 we saw that the loop from Example 2 is non-terminating iff there are $w, x, y, z \in \mathbb{Z}, n_0 \in \mathbb{N}$ such that $p_1^\varphi > 0 \wedge p_2^\varphi > 0$ for all $n > n_0$. We get:*

$$\text{coeffs}\left(p_1^\varphi\right) = \left\{\left(y - \tfrac{1}{3} + \tfrac{1}{2} \cdot w\right)^{(4,0)}, 2^{(1,1)}, \left(x - \tfrac{5}{3}\right)^{(1,0)}\right\}$$
$$\text{coeffs}\left(p_2^\varphi\right) = \left\{\left(\tfrac{2}{3} - 2 \cdot y - w\right)^{(4,0)}, 2^{(1,1)}, \left(x - \tfrac{2}{3}\right)^{(1,0)}\right\}$$

Now it is easy to see that the asymptotic growth of a normalized poly-exponential expression is solely determined by its $\succ$-maximal addend.

Corollary 27 (Maximal Addend Determines Asymptotic Growth). *Let $p \in \mathbb{NPE}$ and let $\max_\succ(\text{coeffs}(p)) = c^{(b,a)}$. Then $\mathcal{O}(p) = \mathcal{O}(c \cdot n^a \cdot b^n)$.*

Proof. Clear, as $c \cdot n^a \cdot b^n$ is the asymptotically dominating addend of p. □

Note that Corollary 27 would be incorrect for the case $c = 0$ if we replaced $\mathcal{O}(p) = \mathcal{O}(c \cdot n^a \cdot b^n)$ with $\mathcal{O}(p) = \mathcal{O}(n^a \cdot b^n)$ as $\mathcal{O}(0) \neq \mathcal{O}(1)$. Building upon Corollary 27, we now show that, for large n, the sign of a normalized poly-exponential expression is solely determined by its $\succ$-maximal coefficient. Here, we define $\text{sign}(c) = -1$ if $c \in \mathbb{Q}_{<0} \cup \{-\infty\}$, $\text{sign}(0) = 0$, and $\text{sign}(c) = 1$ if $c \in \mathbb{Q}_{>0} \cup \{\infty\}$.

Lemma 28 (Sign of $\mathbb{NPE}$s). *Let $p \in \mathbb{NPE}$. Then $\lim_{n\mapsto\infty} p \in \mathbb{Q}$ iff $p \in \mathbb{Q}$ and otherwise, $\lim_{n\mapsto\infty} p \in \{\infty, -\infty\}$. Moreover, we have*

$$\text{sign}\left(\lim_{n\mapsto\infty} p\right) = \text{sign}(\text{unmark}(\max_\succ(\text{coeffs}(p)))).$$

Proof. If $p \notin \mathbb{Q}$, then the limit of each addend of p is in $\{-\infty, \infty\}$ by definition of $\mathbb{NPE}$. As the asymptotically dominating addend determines $\lim_{n\mapsto\infty} p$ and $\text{unmark}(\max_\succ(\text{coeffs}(p)))$ determines the sign of the asymptotically dominating addend, the claim follows. □

Lemma 29 shows the connection between the limit of a normalized poly-exponential expression p and the question whether p is positive for large enough n. The latter corresponds to the existence of a witness for eventual non-termination by Corollary 21 as $\varphi[\overline{x}/\overline{q}_{norm}]$ is a conjunction of inequalities $p > 0$ where $p \in \mathbb{NPE}[\overline{x}]$.

Lemma 29 (Limits and Positivity of $\mathbb{NPE}$s). *Let $p \in \mathbb{NPE}$. Then*

$$\exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ p > 0 \iff \lim_{n\mapsto\infty} p > 0.$$

Proof. By case analysis over $\lim_{n\mapsto\infty} p$. □

Now we show that Corollary 21 allows us to decide eventual non-termination by examining the coefficients of normalized poly-exponential expressions. As these coefficients are in $\mathbb{A}\mathrm{f}[\overline{x}]$, the required reasoning is decidable.

Lemma 30 (Deciding Eventual Positiveness of $\mathbb{NPE}$s). *Validity of*

$$\exists \overline{c} \in \mathbb{Z}^d, n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \textstyle\bigwedge_{i=1}^{k} p_i[\overline{x}/\overline{c}] > 0 \tag{20}$$

where $p_1, \ldots, p_k \in \mathbb{NPE}[\overline{x}]$ is decidable.

Proof. For any p_i with $1 \leq i \leq k$ and any $\overline{c} \in \mathbb{Z}^d$, we have $p_i[\overline{x}/\overline{c}] \in \mathbb{NPE}$. Hence:

$$\begin{aligned}
&\exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \textstyle\bigwedge_{i=1}^{k} p_i[\overline{x}/\overline{c}] > 0 \\
\iff &\textstyle\bigwedge_{i=1}^{k} \exists n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ p_i[\overline{x}/\overline{c}] > 0 \\
\iff &\textstyle\bigwedge_{i=1}^{k} \lim_{n\mapsto\infty} p_i[\overline{x}/\overline{c}] > 0 && \text{(by Lemma 29)} \\
\iff &\textstyle\bigwedge_{i=1}^{k} \mathrm{unmark}(\max_{\succ}(\mathrm{coeffs}(p_i[\overline{x}/\overline{c}]))) > 0 && \text{(by Lemma 28)}
\end{aligned}$$

Let $p \in \mathbb{NPE}[\overline{x}]$ with $\mathrm{coeffs}(p) = \left\{\alpha_1^{(b_1,a_1)}, \ldots, \alpha_\ell^{(b_\ell,a_\ell)}\right\}$ where $\alpha_i^{(b_i,a_i)} \succ \alpha_j^{(b_j,a_j)}$ for all $1 \leq i < j \leq \ell$. If $p[\overline{x}/\overline{c}] = 0$ holds, then $\mathrm{coeffs}(p[\overline{x}/\overline{c}]) = \{0^{(1,0)}\}$ and thus $\mathrm{unmark}(\max_{\succ}(\mathrm{coeffs}(p[\overline{x}/\overline{c}]))) = 0$. Otherwise, there is an $1 \leq j \leq \ell$ with $\mathrm{unmark}(\max_{\succ}(\mathrm{coeffs}(p[\overline{x}/\overline{c}]))) = \alpha_j[\overline{x}/\overline{c}] \neq 0$ and we have $\alpha_i[\overline{x}/\overline{c}] = 0$ for all $1 \leq i \leq j-1$. Hence, $\mathrm{unmark}(\max_{\succ}(\mathrm{coeffs}(p[\overline{x}/\overline{c}]))) > 0$ holds iff $\bigvee_{j=1}^{\ell} \left(\alpha_j[\overline{x}/\overline{c}] > 0 \wedge \bigwedge_{i=0}^{j-1} \alpha_i[\overline{x}/\overline{c}] = 0\right)$ holds, i.e., iff $[\overline{x}/\overline{c}]$ is a model for

$$\mathrm{max\_coeff\_pos}(p) = \textstyle\bigvee_{j=1}^{\ell} \left(\alpha_j > 0 \wedge \bigwedge_{i=0}^{j-1} \alpha_i = 0\right). \tag{21}$$

Hence by the considerations above, (20) is valid iff

$$\exists \overline{c} \in \mathbb{Z}^d.\ \textstyle\bigwedge_{i=1}^{k} \mathrm{max\_coeff\_pos}(p_i)[\overline{x}/\overline{c}] \tag{22}$$

is valid. By multiplying each (in-)equality in (22) with the least common multiple of all denominators, one obtains a first-order formula over the theory of linear integer arithmetic. It is well known that validity of such formulas is decidable. □

Note that (22) is valid iff $\bigwedge_{i=1}^{k} \mathrm{max\_coeff\_pos}(p_i)$ is satisfiable. So to implement our decision procedure, one can use integer programming or SMT solvers to check satisfiability of $\bigwedge_{i=1}^{k} \mathrm{max\_coeff\_pos}(p_i)$. Lemma 30 allows us to prove our main theorem.

Theorem 31. *Termination of triangular loops is decidable.*

Proof. By Theorem 8, termination of triangular loops is decidable iff termination of nnt-loops is decidable. For an nnt-loop (1) we obtain a $\overline{q}_{norm} \in \mathbb{NPE}[\overline{x}]^d$ (see Theorem 17 and Corollary 21) such that (1) is non-terminating iff

$$\exists \overline{c} \in \mathbb{Z}^d, n_0 \in \mathbb{N}.\ \forall n \in \mathbb{N}_{>n_0}.\ \varphi[\overline{x}/\overline{q}_{norm}][\overline{x}/\overline{c}], \tag{20}$$

where φ is a conjunction of inequalities of the form $\alpha > 0$, $\alpha \in \mathbb{Af}[\overline{x}]$. Hence,

$$\varphi[\overline{x}/\overline{q}_{norm}][\overline{x}/\overline{c}] \;=\; \textstyle\bigwedge_{i=1}^{k} p_i[\overline{x}/\overline{c}] > 0$$

where $p_1, \ldots, p_k \in \mathbb{NPE}[\overline{x}]$. Thus, by Lemma 30, validity of (20) is decidable. □

The following algorithm summarizes our decision procedure.

Algorithm 4. Deciding Termination of Triangular Loops

Input: a triangular loop (1)
Result: $\top$ if (1) terminates, $\bot$ otherwise

- apply Definition 3 to (1), i.e.,
  $\varphi \leftarrow \varphi \wedge \varphi[\overline{x}/A\,\overline{x} + \overline{a}]$
  $A \leftarrow A^2$
  $\overline{a} \leftarrow A\,\overline{a} + \overline{a}$
- $\overline{q} \leftarrow$ closed_form$(\overline{x}, A, \overline{a})$ (cf. Algorithm 3)
- compute $\overline{q}_{norm}$ as in Corollary 21
- compute $\varphi[\overline{x}/\overline{q}_{norm}] = \bigwedge_{i=1}^{k} p_i > 0$
- compute $\phi = \bigwedge_{i=1}^{k} \text{max\_coeff\_pos}(p_i)$ (cf. (21))
- **if** ϕ *is satisfiable* **then return** $\bot$ **else return** $\top$

Example 32. *In Example 26 we showed that Example 2 is non-terminating iff*

$$\exists w, x, y, z \in \mathbb{Z},\; n_0 \in \mathbb{N}.\; \forall n \in \mathbb{N}_{>n_0}.\; p_1^{\varphi} > 0 \wedge p_2^{\varphi} > 0$$

is valid. This is the case iff $\text{max\_coeff\_pos}(p_1) \wedge \text{max\_coeff\_pos}(p_2)$*, i.e.,*

$$\begin{array}{c} y - \frac{1}{3} + \frac{1}{2}\cdot w > 0 \vee 2 > 0 \wedge y - \frac{1}{3} + \frac{1}{2}\cdot w = 0 \vee x - \frac{5}{3} > 0 \wedge 2 = 0 \wedge y - \frac{1}{3} + \frac{1}{2}\cdot w = 0 \\ \wedge \\ \frac{2}{3} - 2\cdot y - w > 0 \vee 2 > 0 \wedge \frac{2}{3} - 2\cdot y - w = 0 \vee x - \frac{2}{3} > 0 \wedge 2 = 0 \wedge \frac{2}{3} - 2\cdot y - w = 0 \end{array}$$

is satisfiable. This formula is equivalent to $6 \cdot y - 2 + 3 \cdot w = 0$ *which does not have any integer solutions. Hence, the loop of Example 2 terminates.*

Example 33 shows that our technique does not yield witnesses for non-termination, but it only proves the existence of a witness for *eventual* non-termination. While such a witness can be transformed into a witness for non-termination by applying the loop several times, it is unclear how often the loop needs to be applied.

Example 33. *Consider the following non-terminating loop:*

$$\textbf{while } x > 0 \textbf{ do } \begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x + y \\ 1 \end{bmatrix} \tag{23}$$

The closed form of x *is* $q = [\![n = 0]\!] \cdot x + [\![n \neq 0]\!] \cdot (x + y + n - 1)$*. Replacing* x *with* q_{norm} *in* $x > 0$ *yields* $x + y + n - 1 > 0$*. The maximal marked coefficient of* $x+y+n-1$ *is* $1^{(1,1)}$*. So by Algorithm 4, (23) does not terminate if* $\exists x, y \in \mathbb{Z}.\, 1 > 0$ *is valid. While* $1 > 0$ *is a tautology, (23) terminates if* $x \leq 0$ *or* $x \leq -y$*.*

However, the final formula constructed by Algorithm 4 precisely describes all witnesses for eventual non-termination.

Lemma 34 (Witnessing Eventual Non-Termination). *Let (1) be a triangular loop, let $\overline{q}_{norm}$ be the normalized closed form of (2), and let*

$$(\varphi \wedge \varphi[\overline{x}/A\,\overline{x} + \overline{a}])\,[\overline{x}/\overline{q}_{norm}] = \bigwedge_{i=1}^{k} p_i > 0.$$

Then $\overline{c} \in \mathbb{Z}^d$ witnesses eventual non-termination of (1) iff $[\overline{x}/\overline{c}]$ is a model for

$$\bigwedge_{i=1}^{k} \text{max\_coeff\_pos}(p_i).$$

5 Conclusion

We presented a decision procedure for termination of affine integer loops with triangular update matrices. In this way, we contribute to the ongoing challenge of proving the 15 years old conjecture by Tiwari [15] that termination of affine integer loops is decidable. After linear loops [4], loops with at most 4 variables [14], and loops with diagonalizable update matrices [3,14], triangular loops are the fourth important special case where decidability could be proven.

The key idea of our decision procedure is to compute *closed forms* for the values of the program variables after a symbolic number of iterations n. While these closed forms are rather complex, it turns out that reasoning about first-order formulas over the theory of linear integer arithmetic suffices to analyze their behavior for large n. This allows us to reduce (non-)termination of triangular loops to integer programming. In future work, we plan to investigate generalizations of our approach to other classes of integer loops.

References

1. Bagnara, R., Zaccagnini, A., Zolo, T.: The Automatic Solution of Recurrence Relations. I. Linear Recurrences of Finite Order with Constant Coefficients. Technical report. Quaderno 334. Dipartimento di Matematica, Università di Parma, Italy (2003). http://www.cs.unipr.it/Publications/
2. Ben-Amram, A.M., Genaim, S., Masud, A.N.: On the termination of integer loops. ACM Trans. Programm. Lang. Syst. **34**(4), 16:1–16:24 (2012). https://doi.org/10.1145/2400676.2400679
3. Bozga, M., Iosif, R., Konecný, F.: Deciding conditional termination. Logical Methods Comput. Sci. **10**(3) (2014). https://doi.org/10.2168/LMCS-10(3:8)2014
4. Braverman, M.: Termination of integer linear programs. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 372–385. Springer, Heidelberg (2006). https://doi.org/10.1007/11817963_34
5. Brockschmidt, M., Cook, B., Ishtiaq, S., Khlaaf, H., Piterman, N.: T2: temporal property verification. In: Chechik, M., Raskin, J.-F. (eds.) TACAS 2016. LNCS, vol. 9636, pp. 387–393. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49674-9_22

6. Chen, Y.-F., et al.: Advanced automata-based algorithms for program termination checking. In: Foster, J.S., Grossman, D. (eds.) PLDI 2018, pp. 135–150 (2018). https://doi.org/10.1145/3192366.3192405
7. Chen, H.-Y., David, C., Kroening, D., Schrammel, P., Wachter, B.: Bit-precise procedure-modular termination analysis. ACM Trans. Programm. Lang. Syst. **40**(1), 1:1–1:38 (2018). https://doi.org/10.1145/3121136
8. D'Silva, V., Urban, C.: Conflict-driven conditional termination. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9207, pp. 271–286. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21668-3_16
9. Frohn, F., Giesl, J.: Termination of triangular integer loops is decidable. In: CoRR abs/1905.08664 (2019). https://arxiv.org/abs/1905.08664
10. Frohn, F., Naaf, M., Hensel, J., Brockschmidt, M., Giesl, J.: Lower runtime bounds for integer programs. In: Olivetti, N., Tiwari, A. (eds.) IJCAR 2016. LNCS (LNAI), vol. 9706, pp. 550–567. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40229-1_37
11. Giesl, J., et al.: Analyzing program termination and complexity automatically with AProVE. J. Autom. Reasoning **58**(1), 3–31 (2017). https://doi.org/10.1007/s10817-016-9388-y
12. Larraz, D., Oliveras, A., Rodríguez-Carbonell, E., Rubio, A.: Proving termination of imperative programs using Max-SMT. In: Jobstmann, B., Ray, S. (eds.) FMCAD 2013, pp. 218–225 (2013). https://doi.org/10.1109/FMCAD.2013.6679413
13. Le, T.C., Qin, S., Chin, W.-N.: Termination and non-termination specification inference. In: Grove, D., Blackburn, S. (eds.) PLDI 2015, pp. 489–498 (2015). https://doi.org/10.1145/2737924.2737993
14. Ouaknine, J., Pinto, J.S., Worrell, J.: On termination of integer linear loops. In: Indyk, P. (ed.) SODA 2015, pp. 957–969 (2015). https://doi.org/10.1137/1.9781611973730.65
15. Tiwari, A.: Termination of linear programs. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 70–82. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27813-9_6