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

We show that the question of termination can be reduced to the existential fragment of the first-order theory of $S$. For loops over $\mathbb{R}$, our reduction implies 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-\textit{twn}-loops into \textit{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. Moreover, we formalize a technique to linearize (the updates of) \textit{twn}-loops in our setting and analyze its complexity. Based on these results, we prove complexity bounds for the termination problem of \textit{twn}-loops as well as tight bounds for two important classes of loops which can always be transformed into \textit{twn}-loops.

Finally, we show that there is an important class of linear loops where our decision procedure results in an \textit{efficient} procedure for termination analysis, i.e., where the parameterized complexity of deciding termination is \textit{polynomial}.

Keywords Termination · Polynomial Loops · Decision Procedure · Complexity · Closed Form
1 Introduction

Termination is one of the most important properties of a program. In this work, we study complete approaches for analyzing termination of certain classes of polynomial loops. Compared to incomplete techniques, such approaches have the advantage that they always yield a definite result. In particular, we investigate decidability of termination for our classes of loops.

In the following, we give a short overview on the contributions of our article and highlight how it extends our earlier conference paper [20]. There are already several decidability results for termination of linear loops [8, 10, 19, 30, 41, 44, 48, 61, 65], but only few results on the decidability of termination for certain forms of non-linear loops [42, 43, 46, 63, 64]. Moreover, these previous works only deal with loops whose guards only contain conjunctions, besides [46] which is restricted to guards defining compact sets. In this work, we regard possibly non-linear loops with arbitrary guards, i.e., they may also contain disjunctions and define non-compact sets. More precisely, we consider so-called twin-loops, where the update is mildly restricted to be “triangular” and “weakly non-linear” (see Sect. 2 for a formal definition). We study such loops over \( \mathbb{Z}, \mathbb{Q}, \mathbb{R}_{\text{A}} \) (the real algebraic numbers), and \( \mathbb{R} \), whereas existing decidability results for non-linear loops are restricted to loops over the reals.

Most techniques for proving termination of loops rely on polynomial ranking functions, see, e.g., [1, 5, 6, 7, 9, 50]. However, such ranking functions are only sound for proving termination, i.e., in general, they cannot refute termination. In contrast to ranking functions, we use the computability of closed forms for the iterated update of the loop (Sect. 3). In this way, we can reduce termination of a loop to (in)validity of a certain formula. This reduction, which is a generalization of our earlier results for linear loops over \( \mathbb{Z} \) with conjunctive guards [19], is sound and complete, i.e., validity of the resulting formula proves non-termination, whereas invalidity implies termination. Moreover, our reduction is computable. Analogously to our earlier work for linear loops [19], our decidability results on termination then follow from existing results on the decidability of certain theories. In this way, we show that termination of twin-loops is decidable over \( \mathbb{R}_{\text{A}} \) and \( \mathbb{R} \), and non-termination is semi-decidable over \( \mathbb{Z} \) and \( \mathbb{Q} \) (Sect. 4).

In Sect. 5 we use concepts from algebra to enlarge the classes of loops to which our reduction is applicable. This is done by transforming (certain) non-twin loops into twin-form without affecting their termination behavior (Sect. 5.1). In Sect. 5.2, we discuss for which loops our transformation is applicable. In this way, we generalize our results to a broader class of polynomial loops (Sect. 5.3).

In contrast to our earlier conference paper [20], in Sect. 6 we formalize the technique of [47] to linearize (the updates of) twin-loops in our setting. Using this formalization, we develop novel results on the complexity of linearization. Afterwards, based on our decision procedure for termination in Sect. 4, on the transformation of Sect. 5, and on our complexity results for linearization from Sect. 6, we study the complexity of deciding termination in Sect. 7.
In Sect. 7.1 we show that deciding termination of linear loops with rational spectrum over \( \mathbb{Z}, \mathbb{Q}, \mathbb{R}_A, \) and \( \mathbb{R} \) is \( \text{Co-NP} \)-complete. Moreover, we show that deciding termination of linear-update loops (where the update is linear but the guard may be non-linear) with real spectrum over \( \mathbb{R}_A \) and \( \mathbb{R} \) is \( \forall \mathbb{R} \)-complete. Here, a loop has \textit{rational} or \textit{real spectrum}, respectively, if its update matrix has rational or real eigenvalues only, and \( \forall \mathbb{R} \) is the complexity class of problems which can be reduced to validity of a universally quantified formula of polynomial inequations over the reals. We also analyze the complexity of deciding termination for arbitrary \( \text{twn} \)-loops (with possibly non-linear updates) in Sect. 7.2. In our conference paper [20, Thm. 45], we had only analyzed this case for a \textit{bounded} number of variables. In contrast, we now extend our analysis to the general case where the number of variables is \textit{not} restricted (Thm. 7.6). To this end, we need our new results from Sect. 6 on the complexity of linearizing \( \text{twn} \)-loops.

Finally, in contrast to [20], we identify a class of linear loops (\textit{uniform loops}) where termination can be interpreted as a parameterized problem which is decidable in polynomial time when fixing such a parameter (Sect. 7.3). Based on the transformation of Sect. 5, we show that the closed forms arising from uniform loops have a special structure. Therefore, here (in)validity of the formula from Sect. 4 which encodes termination can be checked in polynomial time.

Related work is discussed in Sect. 8 and all missing proofs can be found in App. A. So the current paper extends [20] by the following new material:

- Sect. 6 on the linearization of \( \text{twn} \)-loops and its complexity.
- Thm. 7.6 on the complexity of deciding termination for arbitrary \( \text{twn} \)-loops where the number of variables is \textit{not} restricted.
- Sect. 7.3 on \textit{uniform loops} where the parameterized complexity of deciding termination is \textit{polynomial}.
- Several additional explanations and remarks.

## 2 Preliminaries

A \textit{(polynomial) loop over a ring} \( S \) has the form in Fig. 1, where \( \mathbb{Z} \leq S \leq \mathbb{R}_A \) and \( \leq \) denotes the subring relation. Here, \( \vec{x} \) is a vector of \( d \geq 1 \) pairwise different variables that range over \( S \) and \( \vec{u} \in (S[\vec{x}])^d \) where \( S[\vec{x}] \) is the set of polynomials over \( \vec{x} \) with coefficients in \( S \). To improve readability, we use row-and column-vectors interchangeably. The guard \( \varphi \) is an arbitrary propositional (i.e., quantifier-free) formula over the atoms \( \{ p \succ 0 \mid p \in S[\vec{x}], \succ \in \{ \geq, > \} \} \). We denote the set of all such formulas by \( \text{Th}_{qf}(S) \). In our setting, negation is syntactic sugar as, e.g., \( \neg(p > 0) \) is equivalent to \( \neg p \geq 0 \). So w.l.o.g. the guard (or condition) \( \varphi \) of a loop is built from atoms, \( \land \), and \( \lor \).

We require \( S \leq \mathbb{R}_A \) instead of \( S \leq \mathbb{R} \), as it is unclear how to represent transcendental numbers on computers. However, in Sect. 4 we will see that the loops considered in this work terminate over \( \mathbb{R} \) iff they terminate over \( \mathbb{R}_A \). Thus, our results immediately carry over to loops where the variables range...
over \( \mathbb{R} \). Hence, we sometimes also consider loops over \( \mathcal{S} = \mathbb{R} \). However, even then we restrict ourselves to loops where all constants in \( \varphi \) and \( \vec{u} \) are algebraic.

We often represent a loop as in Fig. 1 by the tuple \( (\varphi, \vec{u}) \) of the condition \( \varphi \) and the update \( \vec{u} = (u_1, \ldots, u_d) \). Unless stated otherwise, \( (\varphi, \vec{u}) \) is always a loop over \( \mathcal{S} \) using the variables \( \vec{x} = (x_1, \ldots, x_d) \) where \( \mathbb{Z} \leq \mathcal{S} \leq \mathbb{R}_\mathcal{A} \). We use the following notions for certain classes of loops: A \emph{linear-update loop} has the form \( (\varphi, A \cdot \vec{x} + \vec{b}) \), and it has \emph{rational} or \emph{real spectrum}, respectively, if \( A \) has rational or real eigenvalues only. Then all eigenvalues of \( A \) are real algebraic numbers, since \( \vec{u} \in (\mathbb{R}[\vec{x}])^d \) implies that all constants in \( \vec{u} \) (and thus in \( A \)) are algebraic. A \emph{linear loop} is a linear-update loop where \( \varphi \) is linear (i.e., its atoms are only linear inequations). Here, “linear” refers to “linear polynomials”, i.e., of degree at most 1, so it includes affine functions. A \emph{conjunctive loop} is a loop \( (\varphi, \vec{u}) \) where \( \varphi \) does not contain disjunctions.

For any entity \( s \), \( s[x/t] \) results from \( s \) by replacing all free occurrences of \( x \) by \( t \). Similarly, if \( \vec{x} = (x_1, \ldots, x_d) \) and \( \vec{t} = (t_1, \ldots, t_d) \), then \( s[\vec{x}/\vec{t}] \) results from \( s \) by replacing all free occurrences of \( x_i \) by \( t_i \), for each \( 1 \leq i \leq d \).

Any vector of polynomials \( \vec{u} \in (\mathcal{S}[\vec{x}])^d \) can also be regarded as a function \( \vec{u} : (\mathcal{S}[\vec{x}])^d \to (\mathcal{S}[\vec{x}])^d \) where for any \( \vec{p} \in (\mathcal{S}[\vec{x}])^d \), \( \vec{u}(\vec{p}) = \vec{u}[\vec{x}/\vec{p}] \) results from \emph{applying} the polynomials \( \vec{u} \) to the polynomials \( \vec{p} \). Similarly, we can also apply a formula to polynomials \( \vec{p} \in (\mathcal{S}[\vec{x}])^d \). To this end, we define \( \psi(\vec{p}) = \psi[\vec{x}/\vec{p}] \) for first-order formulas \( \psi \) with the free variables \( \vec{x} \). As usual, function application associates to the left, i.e., \( \vec{u}(\vec{b})(\vec{p}) \) stands for \( (\vec{u}(\vec{b}))(\vec{p}) \). However, obviously \( (\vec{u}(\vec{b}))(\vec{p}) = \vec{u}(\vec{b}[\vec{p}]) \) since applying polynomials only means that one instantiates variables.

Def. 2.1 formalizes the intuitive notion of termination for a loop \( (\varphi, \vec{u}) \) and the related notion of \emph{eventual termination} \([10, 61]\). Here, \( \vec{u}^n \) denotes the \( n \)-fold application of \( \vec{u} \), i.e., \( \vec{u}^n(\vec{e}) = \vec{e} \) and \( \vec{u}^{n+1}(\vec{e}) = \vec{u}(\vec{u}^n(\vec{e})) \).

\textbf{Definition 2.1} (Termination). Let \( (\varphi, \vec{u}) \) be a loop over \( \mathcal{S} \) and \( \vec{e} \in \mathcal{S}^d \). If \( \forall n \in \mathbb{N}, \varphi(\vec{u}^n(\vec{e})) \) holds, then \( \vec{e} \) is a \emph{witness for non-termination}. If \( (\varphi, \vec{u}) \) does not have any witnesses for non-termination, it \emph{terminates} (over \( \mathcal{S} \)).

If \( \vec{u}^n(\vec{e}) \) is a witness for non-termination for some \( n_0 \in \mathbb{N} \), then \( \vec{e} \) is called a \emph{witness for eventual non-termination}.

\( E \) \( NT_{(\varphi, \vec{u})} \) denotes the set of witnesses for (eventual) non-termination of \( (\varphi, \vec{u}) \) and we define \( T_{(\varphi, \vec{u})} = \mathcal{S}^d \setminus NT_{(\varphi, \vec{u})} \).

For any entity \( s \), \( \mathcal{V}(s) \) is the set of all its free variables. Given an assignment \( \vec{x} \leftarrow \vec{u} \), the relation \( \succ \in \mathcal{V}(\vec{u}) \times \mathcal{V}(\vec{u}) \) is the transitive closure of \( \{(x_i, x_j) | i, j \in \{1, \ldots, d\}, i \neq j, x_j \in \mathcal{V}(u_i)\} \), i.e., \( x_i \succ x_j \) means that \( x_i \) depends on \( x_j \). For example, if \( \vec{u} = (x_1 + x_2, x_2 + 1) \) then we have \( \succ = \{(x_1, x_2)\} \).

Now we can introduce the class of \emph{twin-loops}. A loop \( (\varphi, \vec{u}) \) is \emph{triangular} if \( \succ \) is well founded. It is \emph{weakly non-linear} if there is no \( 1 \leq i \leq d \) such that \( x_i \)}
occurs in a non-linear monomial of \( u_i \), i.e., \( u_i = c_i \cdot x_i + p_i \) where \( c_i \in S \) and \( p_i \in S[\vec{x}] \) does not contain \( x_i \). A \textit{tnn-loop} is triangular and weakly non-linear.

We call a loop \textit{non-negative} if it is weakly non-linear and the coefficient \( c_i \) of the monomial \( x_i \) in \( u_i \) is non-negative for all \( 1 \leq i \leq d \). A \textit{tnn-loop} is triangular and non-negative, i.e., a \textit{tnn-loop} is a special form of a \textit{tnw-loop}.

The restriction to triangular loops prohibits “cyclic dependencies” of variables (e.g., where the new values of \( x_1 \) and \( x_2 \) both depend on the old values of \( x_1 \) and \( x_2 \)). For example, the loop whose body consists of the assignment \( (x_1, x_2) \leftarrow (x_1 + x_2^2, x_2 + 1) \) is triangular since \( \succ = \{(x_1, x_2)\} \) is well founded, whereas a loop with the body \( (x_1, x_2) \leftarrow (x_1 + x_2^2, x_2 + 1) \) is not triangular.

By the restriction to \textit{tnw}-loops we can compute a closed form for the \textit{n}-fold application of the update \( \vec{u} \) by handling one variable after the other. A vector \( \vec{q} \) of \( d \) arithmetic expressions over \( \vec{x} \) and a distinguished variable \( n \) is a \textit{closed form} for \( \vec{u} \in (S[\vec{x}])^d \) if \( \vec{q}[\vec{x}/\vec{e}, n/n'] = \vec{u}^n(\vec{e}) \) for all \( \vec{e} \in S^d \) and \( n' \in \mathbb{N} \), i.e., both vectors of expressions evaluate to the same element of \( S^d \). Thus, \( \vec{q} = \vec{u}^{n} \).

Triangular loops are very common in practice. For example, in [21], 1511 polynomial loops were extracted from the \textit{Termination Problems Data Base (TPDB)} [62], the benchmark collection used at the annual \textit{Termination and Complexity Competition} [25], and only 26 of them were non-triangular.

A loop with the body \((x_1, x_2) \leftarrow (x_1 + x_2^2, x_2 + 1)\) is weakly non-linear, while a loop with \((x_1, x_2) \leftarrow (x_1 \cdot x_2, x_2 + 1)\) is not. In particular, weak non-linearity excludes assignments like \( x_1 \leftarrow x_1^2 \) that need exponential space, as \( x_1 \) grows doubly exponentially. By permuting variables, the update of every \textit{tnw-loop} can be transformed to the following form where \( c_i \in S \) and \( p_i \in S[\vec{x}_{i+1}, \ldots, \vec{x}_d] \):

\[
\begin{align*}
x_1 & \leftarrow c_1 \cdot x_1 + p_1 \\
\cdots \\
x_d & \leftarrow c_d \cdot x_d + p_d
\end{align*}
\]

\textbf{Example 2.2.} Consider the loop \( L_{\text{ex}} \) over the ring \( \mathbb{Z} \) in Fig. 2. This loop is triangular since \( \succ_{\vec{u}} = \{(x_1, x_2), (x_1, x_3), (x_2, x_3)\} \) is well founded. Moreover, it is weakly non-linear. Since the coefficient of \( x_i \) is 1 in the update of \( x_i \) for all \( 1 \leq i \leq 3 \), this loop is also non-negative, i.e., \( L_{\text{ex}} \) is \textit{tnn}.

Our \textit{tnw-loops} are a special case of \textit{solvable loops}.

\textbf{Definition 2.3 (Solvable Loops [55]).} A loop \( (\varphi, \vec{u}) \) is \textit{solvable} if there is a partitioning \( \mathcal{J} = \{J_1, \ldots, J_k\} \) of \( \{1, \ldots, d\} \) such that for each \( 1 \leq i \leq k \) we have \( \vec{u}_{J_i} = A_i \cdot \vec{x}_{J_i} + \vec{p}_i \), where \( \vec{u}_{J_i} \) is the vector of all \( u_j \) with \( j \in J_i \) (and \( \vec{x}_{J_i} \) is defined analogously), \( A_i = \{J_i\} \), \( A_i \in S^{d \times d} \), and \( \vec{p}_i \in (S[\vec{x}_{J_{i+1}}, \ldots, \vec{x}_{J_d}])^d \).

The eigenvalues of a solvable loop are the union of the eigenvalues of all \( A_i \).

So solvable loops allow for blocks of variables with linear dependencies, and \textit{tnw}-loops correspond to the case that all blocks have size 1. While our approach could be generalized to solvable loops with real eigenvalues, Thm. 5.15 (Sect. 5) shows that this generalization does not increase its applicability.

For our decidability results in Sect. 4, we reduce termination to the \textit{existential fragment} \( \text{Th}_3(S) \) of the first-order theory of \( S \) (see, e.g., [51, 56]). \( \text{Th}_3(S) \) consists of all formulas \( \exists \vec{y} \in S^k \cdot \varphi \) where \( k \in \mathbb{N} \) and the propositional formula \( \varphi \) is built from \( \land \) and \( \lor \) over the atoms \( \{p \triangleright 0 \mid p \in \mathbb{Q}[\vec{y}, \vec{z}], \triangleright \in \{\geq, >\}\} \).

\textbf{Termination of Triangular Polynomial Loops 5}
Here, \( \vec{y} \) and \( \vec{z} \) are pairwise disjoint vectors of variables. The free variables \( \vec{z} \) range over \( \mathbb{R} \) and they are needed in Sect. 5 to characterize subsets of real (algebraic) numbers by formulas.

The existential fragment of the first-order theory of \( S \) and \( \mathbb{R}_A \) is the set \( \text{Th}_S(S, \mathbb{R}_A) \) of all formulas \( \exists \vec{y}' \in \mathbb{R}_A^{k'}, \vec{y} \in S^{k}, \psi \), with a propositional formula \( \psi \) over \( \{ p > 0 \mid p \in \mathbb{Q}[\vec{y}', \vec{y}, \vec{z}], \triangleright \in \{\geq, >\} \} \) where \( k', k \in \mathbb{N} \) and the variables \( \vec{y}', \vec{y}, \vec{z} \) are pairwise disjoint. A formula without free variables is closed.

In the following, we also consider formulas over inequations \( p \triangleright 0 \) where \( p \)'s coefficients are from \( \mathbb{R}_A \) to be elements of \( \text{Th}_S(\mathbb{R}_A) \) (resp. \( \text{Th}_S(S, \mathbb{R}_A) \)). The reason is that real algebraic numbers are \( \text{Th}_S(\mathbb{R}_A) \)-definable.

Validity of closed formulas from \( \text{Th}_S(S) \) or \( \text{Th}_S(S, \mathbb{R}_A) \) is decidable if \( S \in \{ \mathbb{R}_A, \mathbb{R} \} \) and semi-decidable if \( S \in \{ \mathbb{Z}, \mathbb{Q} \} \) [13, 60]. By undecidability of Hilbert’s Tenth Problem [45], it is undecidable for \( S = \mathbb{Z} \). While validity of full first-order formulas (i.e., also containing universal quantifiers) over \( S = \mathbb{Q} \) is undecidable [52], it is still open whether validity of closed formulas from \( \text{Th}_S(\mathbb{Q}) \) or \( \text{Th}_S(\mathbb{Q}, \mathbb{R}_A) \) is decidable. However, validity of linear closed formulas from \( \text{Th}_S(S) \) or \( \text{Th}_S(S, \mathbb{R}_A) \) is decidable for all \( S \in \{ \mathbb{Z}, \mathbb{Q}, \mathbb{R}_A, \mathbb{R} \} \) [15, 26, 32, 36]. Here, a formula is linear if it only contains atoms \( p \triangleright 0 \) where \( p \) is linear.

3 Reducing Termination of twn-Loops to Termination of tnn-Loops

For analyzing termination of twn-loops, we can restrict ourselves to tnn-loops, as twn-loops can be (automatically) transformed into tnn-loops via chaining.

**Definition 3.1** (Chaining). Chaining a loop \( (\varphi, \vec{u}) \) yields \( (\varphi \land \varphi(\vec{u}), \bar{u}(\vec{u})) \).

So for example, chaining the loop \( (x_1 > 0, -x_1) \) (i.e., “while \( x_1 > 0 \) do \( x_1 \leftarrow -x_1 \)” yields \( (x_1 > 0 \land -x_1 > 0, x_1) \). Analogous to [19] where chaining was used for triangular linear loops, we obtain the following theorem.

**Theorem 3.2** (Soundness of Chaining). Let \((\varphi, \vec{u})\) be a twn-loop on \( S^d \).

(a) \((\varphi \land \varphi(\vec{u}), \bar{u}(\vec{u}))\) is tnn, i.e., it is triangular and the coefficient of each \( x_i \) in \( \bar{u}(\vec{u}) \) is non-negative.

(b) \((\varphi, \vec{u})\) terminates on \( \bar{e} \in S^d \) iff \((\varphi \land \varphi(\vec{u}), \bar{u}(\vec{u}))\) terminates on \( \bar{e} \).

As chaining is clearly computable, we get the following corollary.

**Corollary 3.3.** Termination of twn-loops is reducible to termination of tnn-loops.

It is well known that closed forms for twn-loops are computable (see, e.g., [34, 66]) since their bodies correspond to \( C\)-finite recurrences, which are known to be solvable [33]. The resulting closed forms may contain polynomial arithmetic and exponentiation w.r.t. \( n \) (as, e.g., \( x_1 \leftarrow 2 \cdot x_1 \) has the closed form \( x_1 \cdot 2^n \)) as well as certain piecewise defined functions. For example, the closed form \( x_1^{(n)} \) of \( x_1 \leftarrow 1 \) is \( x_1^{(0)} = x_1 \) and \( x_1^{(n)} = 1 \) for all \( n \in \mathbb{N} \) with \( n > 0 \).
We represent closed forms using poly-exponential expressions [19], where instead of handling piecewise defined functions via disjunctions (as in [34]), we simulate them via Iverson brackets. For a formula \( \psi \) over \( n \), its \textit{Iverson bracket} \( [\psi] : \mathbb{N} \to \{0,1\} \) evaluates to 1 iff \( \psi \) is satisfied (i.e., \( [\psi](e) = 1 \) if \( \psi[n/e] \) holds and \( [\psi](e) = 0 \), otherwise). Later, Iverson brackets can be replaced by the constants 0 or 1, as we only use them for formulas \( \psi \) that are constantly false or true for large enough values of \( n \), see Sect. 4. Poly-exponential expressions are sums of arithmetic terms over the variables \( \vec{x} \) and the additional designated variable \( n \), where it is always clear which addend determines the asymptotic growth of the expression when increasing \( n \). This is crucial for our reducibility proof in Sect. 4. Def. 3.4 slightly generalizes the poly-exponential expressions from [19, Def. 9] by allowing arbitrary polynomials over \( \vec{x} \) (instead of just linear expressions) as coefficients. In the following, for any set \( X \subseteq \mathbb{R} \), any \( k \in X \), and \( b \in \{\geq, >\} \), let \( X_{b,k} = \{x \in X \mid x b k \} \).

**Definition 3.4 (Poly-Exponential Expressions).** Let \( C \) be the set of all finite conjunctions over \( \{n = c, n \neq c \mid c \in \mathbb{N}\} \) where \( n \) is a designated variable and let \( Q_S = \{\frac{r}{s} \mid r \in S, s \in S_{>0}\} \) be the quotient field of \( S \). Then the set of all poly-exponential expressions with the variables \( \vec{x} \) over \( S \) is

\[
\text{PE}_S[\vec{x}] = \{\sum_{j=1}^r [\psi_j] \cdot a_j \cdot n^{\alpha_j} \cdot b_j^n \mid \ell, a_j \in \mathbb{N}, \psi_j \in C, \alpha_j \in Q_S[\vec{x}], b_j \in S_{>0}\}.
\]

An example for a poly-exponential expression over \( Z \) (with \( Q_Z = Q \)) is

\[
[n \neq 0 \land n \neq 1] \cdot \left(\frac{1}{2} \cdot x_1^2 + \frac{3}{4} \cdot x_2 - 1\right) \cdot n^3 \cdot 3^n + [n = 1] \cdot (x_1 - x_2).
\]

The restriction to \( tnn \)-loops ensures that for the closed form \( \vec{q} \) of the update we indeed have \( \vec{q} \in (\text{PE}_S[\vec{x}])^d \). For example, for arbitrary matrices \( A \in \mathbb{R}_{d \times d}^d \), the update \( \vec{x} \leftarrow A \cdot \vec{x} \) is known to admit a closed form as in Def. 3.4 with complex \( b_j \)'s, whereas real numbers suffice for triangular matrices. Moreover, non-negativity is required to ensure \( b_j > 0 \) (e.g., a non-\( tnn \) loop with the update \( x_1 \leftarrow -x_1 \) has the closed form \( x_1 \cdot (-1)^n \)).

**Example 3.5.** For \( L_{xz} \) in Fig. 2, the closed form is \( \vec{q} = (\text{PE}_Z[\vec{x}])^3 \) with

\[
\vec{q} = 4 \cdot x_3^3 \cdot n^3 + (-2 \cdot x_3^2 - 2 \cdot x_2 \cdot x_3^3) \cdot n^2 + (x_2^2 \cdot x_3 + x_2^2 \cdot x_3 + 2 \cdot x_2 \cdot x_3) \cdot n + x_1^4.
\]

4 Reducing Termination of \( tnn \)-Loops to \( \text{Th}_3(S) \)

It is known that the bodies of \( tnn \)-loops can be linearized [47], i.e., one can reduce termination of a \( tnn \)-loop \( \langle \varphi, \vec{u} \rangle \) to termination of a linear-update \( tnn \)-loop \( \langle \varphi', \vec{u}' \rangle \), where \( \varphi' \) may be non-linear. See Sect. 6 for a discussion of linearization and novel results on the linearization procedure. Moreover, [63, 64] showed decidability of termination for certain classes of conjunctive linear-update loops over \( \mathbb{R} \), which cover conjunctive linear-update \( tnn \)-loops. So, by combining the results of [47] and [63, 64], one can conclude that termination of \( \text{conjunctive tnn} \)-loops over \( \mathbb{R} \) is decidable.
In contrast, we now present a reduction of termination of $tnn$-loops to $Th_{\mathbb{Z}}(S)$ which applies to $tnn$-loops over any ring $\mathbb{Z} \leq S \leq \mathbb{R}$ and which can also handle disjunctions in the loop condition. Moreover, our reduction yields tight complexity results on termination of linear loops over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and on termination of linear-update loops over $\mathbb{R}^n$ and $\mathbb{R}$, and on termination of linear-update loops over $\mathbb{R}^n$ and $\mathbb{R}$ (see Sect. 7.1).

Similar to [19], our reduction exploits that for $tnn$-loops $(\varphi, \vec{u})$ there is a closed form $\vec{q}$ for $\vec{u}$ with $\vec{q} \in (\mathbb{PE}_S[\vec{x}])^d$. However, in [19] we only considered conjunctive linear loops over $\mathbb{Z}$. In contrast, we now analyze loops over $S$ for any $\mathbb{Z} \leq S \leq \mathbb{R}$ and allow non-linearity and arbitrary propositional formulas in the condition. Thus, the correctness proofs differ substantially from [19].

More precisely, we show that there is a function with the following specification that is computable in polynomial time:

**Input:** a $tnn$-loop $(\varphi, \vec{u})$ over $S$ with closed form $\vec{q} \in (\mathbb{PE}_S[\vec{x}])^d$

**Result:** a closed formula $\chi \in Th_{\mathbb{Z}}(S)$ such that

$$\chi \text{ is valid iff } (\varphi, \vec{u}) \text{ does not terminate on } S^d$$

We use the concept of eventual non-termination, i.e., the condition of the loop may be violated finitely often, see Def. 2.1. Clearly, $(\varphi, \vec{u})$ is non-terminating iff it is eventually non-terminating [48]. The formula $\chi$ in (1) will encode the existence of a witness for eventual non-termination. By the definition of $\vec{q}$, eventual non-termination of $(\varphi, \vec{u})$ on $S^d$ is equivalent to validity of

$$\exists \vec{x} \in S^d, \ n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{>n_0}. \ \varphi(\vec{q}).$$

**Example 4.1.** Continuing Ex. 3.5, $L_{ex}$ is eventually non-terminating over $\mathbb{Z}$ iff there is a corresponding witness $\vec{e} \in \mathbb{Z}^3$, i.e., iff

$$\exists x_1, x_2, x_3 \in \mathbb{Z}, \ n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{>n_0}, \ p > 0$$

is valid where

$$p = (x_1 + x_2^3) + (x_2^2 \cdot x_3 + \frac{2}{3} \cdot x_3^5 + 2 \cdot x_2 \cdot x_3^3 - 4 \cdot x_2 \cdot x_3^2) \cdot n + (-2 \cdot x_3^5 - 2 \cdot x_2 \cdot x_3^3 + 4 \cdot x_3^4) \cdot n^2 + (\frac{4}{3} \cdot x_3^5) \cdot n^3.$$

Let $\vec{q}_{\text{norm}}$ be like $\vec{q}$, but each factor $[\psi]$ is replaced by 0 if it contains an equation and by 1, otherwise. The reason is that for large enough $n$, equations in $\psi$ become false and negated equations become true. So (2) is equivalent to

$$\exists \vec{x} \in S^d, \ n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{>n_0}. \ \varphi(\vec{q}_{\text{norm}}).$$

In this way, we obtain normalized poly-exponential expressions.

**Definition 4.2 (Normalized $\mathbb{PE}$s).** We call $p \in \mathbb{PE}_S[\vec{x}]$ normalized if it is in

$$\mathbb{NP}_{\mathbb{PE}}[\vec{x}] = \{ \sum_{j=1}^{t} \alpha_j \cdot n^{a_j} \cdot b_j^n \mid t, a_j \in \mathbb{N}, \ \alpha_j \in \mathbb{Q}[\vec{x}], \ b_j \in S_{>0} \},$$

where w.l.o.g. $(b_i, a_i) \neq (b_j, a_j)$ if $i \neq j$. We define $\mathbb{NP}_{\mathbb{PE}} = \mathbb{NP}_{\mathbb{PE}}[\emptyset]$.

As $\varphi$ is a propositional formula over $S[\vec{x}]$-inequations, $\varphi(\vec{q}_{\text{norm}})$ is a propositional formula over $\mathbb{NP}_{\mathbb{PE}}[\vec{x}]$-inequations. By (4), we need to check whether $\exists \vec{x} \in S^d. \ \varphi(\vec{q}_{\text{norm}})$ is valid for large enough $n$. To this end, we will examine the dominant terms in the inequations of $\varphi(\vec{q}_{\text{norm}})$.
Definition 4.3 (Asymptotic Domination [37]). A function \( g : \mathbb{N} \to \mathbb{R} \) dominates a function \( f : \mathbb{N} \to \mathbb{R} \) asymptotically (\( f \in o(g) \)) if for all \( m > 0 \) there is an \( n_0 \in \mathbb{N} \) such that\(^1\) \( |f(n)| < m \cdot |g(n)| \) for all \( n \in \mathbb{N}_{>n_0} \).

Now we can state the following lemma which generalizes [19, Lemma 24].

Lemma 4.4. Let \( b_1, b_2 \in S_{>0} \) and \( a_1, a_2 \in \mathbb{N} \). If \( (b_2, a_2) \succ_{lex} (b_1, a_1) \), then \( n^{a_1} \cdot b_1^j \in o(n^{a_2} \cdot b_2^j) \), where \((b_2, a_2) \succ_{lex} (b_1, a_1)\) iff \( b_2 > b_1 \) or \( b_2 = b_1 \land a_2 > a_1 \).

In the following, let \( p \geq 0 \) or \( p > 0 \) occur in \( \varphi(\mathbf{a}_{norm}) \). We can order the coefficients of \( p \) according to the asymptotic growth of their addends w.r.t. \( n \).

Definition 4.5 (Ordering Coefficients). Marked coefficients are of the form \( \alpha(b, a) \) where \( \alpha \in \mathbb{Q}_S[\mathbf{a}], b \in S_{>0}, \) and \( a \in \mathbb{N} \). We define unlmark(\( \alpha(b, a) \)) \( = \alpha \) and \( \alpha(b_2, a_2) \succ_{coeff} \alpha(b_1, a_1) \) if \( (b_2, a_2) \succ_{lex} (b_1, a_1) \). Let \( p = \sum_{j=1}^{\ell} \alpha_j \cdot n^{a_j} \cdot b_j^j \in NPE_S[\mathbf{a}], \) where \( \alpha_j \neq 0 \) for all \( 1 \leq j \leq \ell \). Then the marked coefficients of \( p \) are

\[
\text{coeffs}(p) = \begin{cases} \{0^{(1,0)}\} & \text{if } \ell = 0 \\ \{\alpha_j^{(b_j, a_j)} | 1 \leq j \leq \ell\} & \text{otherwise.} \end{cases}
\]

Example 4.6. Continuing Ex. 4.1, \( \text{coeffs}(p) \) is \( \{\alpha_1^{(1,0)}, \alpha_2^{(1,1)}, \alpha_3^{(1,2)}, \alpha_4^{(1,3)}\} \) where

\[
\begin{align*}
\alpha_1 &= x_1 + x_2^2 \\
\alpha_2 &= x_2^3 \cdot x_3 + \frac{2}{3} \cdot x_3^5 + 2 \cdot x_2 \cdot x_3^3 - 4 \cdot x_2 \cdot x_3^2 \\
\alpha_3 &= -2 \cdot x_3^2 - 2 \cdot x_2 \cdot x_3^4 + 4 \cdot x_3^4 \\
\alpha_4 &= \frac{4}{3} \cdot x_3^3 
\end{align*}
\]

Note that \( p(\mathbf{c}) \in NPE_S \) for any \( \mathbf{c} \in S^d \), i.e., the only variable in \( p(\mathbf{c}) \) is \( n \). Now the \( \succ_{coeff} \)-maximal addend determines the asymptotic growth of \( p(\mathbf{c}) \):

\[
o(p(\mathbf{c})) = o(k \cdot n^a \cdot b^a) \quad \text{where } k(b, a) = \max_{\succ_{coeff}} (\text{coeffs}(p(\mathbf{c}))).
\]

Note that (5) would be incorrect for the case \( k = 0 \) if we replaced \( o(p(\mathbf{c})) \) \( = o(k \cdot n^a \cdot b^a) \) with \( o(p(\mathbf{c})) \) \( = o(n^a \cdot b^a) \) as \( o(0) = 0 \neq o(1) \). Obviously, (5) implies

\[
\exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. \ sign(p(\mathbf{c})) = sign(k),
\]

where \( \text{sign}(0) = 0, \text{sign}(k) = 1 \) if \( k > 0 \), and \( \text{sign}(k) = -1 \) if \( k < 0 \). This allows us to reduce eventual non-termination to \( \text{Th}_\exists(S) \) if \( \varphi \) is an atom. In the following, let \( \text{coeffs}(p) = \{\alpha_1^{(b_1, a_1)}, \ldots, \alpha_\ell^{(b_\ell, a_\ell)}\} \), where \( \alpha_j^{(b_j, a_j)} \succ_{coeff} \alpha_j^{(b_j, a_j)} \) for all \( 1 \leq i < j \leq \ell \). Then we define

\[
\text{red}(p > 0) = \bigwedge_{i=1}^\ell (\alpha_i > 0 \land \bigwedge_{j=i+1}^\ell \alpha_j = 0)
\]

and

\[
\text{red}(p \geq 0) = \text{red}(p > 0) \lor \bigwedge_{j=1}^\ell \alpha_i = 0.
\]

\(^1\) Our definition is slightly more general than the original definition of [37] (which requires \( \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \)), but both definitions are equivalent if \( g(n) \) is positive for large enough \( n \).
**Lemma 4.7.** Given \( p \in \mathbb{NPE}_S[\vec{x}] \) and \( \triangleright \in \{\geq, >\} \), validity of

\[
\exists \vec{x} \in S^d, \ n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{n_0}. \ p \triangleright 0
\]

(8)

can be reduced to validity of the closed formula \( \exists \vec{x} \in S^d. \ \text{red}(p \triangleright 0) \) from \( \text{Th}_3(S) \). This reduction takes polynomially many steps in the size of \( p \).

To gain intuition for the formula \( \text{red}(p \triangleright 0) \), note that by (6), we have \( p(\vec{c}) > 0 \) for large enough values of \( n \) iff the coefficient of the asymptotically fastest growing addend \( a(\vec{c}) \cdot n^a \cdot b^n \) that does not vanish (i.e., where \( a(\vec{c}) \neq 0 \)) is positive. Similarly, we have \( p(\vec{e}) < 0 \) for large enough \( n \) iff \( a(\vec{e}) < 0 \). If all addends of \( p \) vanish when instantiating \( \vec{x} \) with \( \vec{c} \), then \( p(\vec{c}) = 0 \). In other words, (8) holds iff there is an \( \vec{c} \in S^d \) such that unmark \( \{\max_{x \in S} (\text{coefs}(p(\vec{e}))) \} \triangleright 0 \). The formula \( \text{red}(p \triangleright 0) \) expresses the latter in \( \text{Th}_3(S) \).

**Example 4.8.** We continue Ex. 4.6. By the construction in Lemma 4.7, (3) is valid iff \( \exists x_1, x_2, x_3 \in \mathbb{Z}. \ \text{red}(p \triangleright 0) \) is valid, where \( \text{red}(p \triangleright 0) \) is

\[
(\alpha_1 > 0 \wedge \alpha_2 = 0 \wedge \alpha_3 = 0 \wedge \alpha_4 = 0) \lor (\alpha_2 > 0 \wedge \alpha_3 = 0 \wedge \alpha_4 = 0) \\
\lor (\alpha_3 > 0 \wedge \alpha_4 = 0) \lor \alpha_4 > 0.
\]

For example, \([x_1/ -4, x_2/2, x_3/1]\) satisfies \( \alpha_4 > 0 \) as \((\frac{4}{3} \cdot x_3^5) [x_1/ -4, x_2/2, x_3/1] > 0 \). Thus, \((-4, 2, 1)\) witnesses eventual non-termination of \( L_{\text{ex}} \) over \( \mathbb{Z} \).

Now we lift our reduction to propositional formulas. To handle disjunctions, the proof of Thm. 4.9 exploits the crucial insight that a \( \text{npe}-\)loop \( (\varphi \lor \varphi', \vec{u}) \) terminates iff \( \langle \varphi, \vec{u} \rangle \) and \( \langle \varphi', \vec{u} \rangle \) terminate, which is not true in general (as, e.g., witnessed by the loop \( (x_1 > 0 \lor -x_1 > 0, -x_1) \)). In the following, the formula \( \text{red}(\xi) \) results from \( \xi \) by replacing each atom \( p \triangleright 0 \) in \( \xi \) by \( \text{red}(p \triangleright 0) \).

**Theorem 4.9** (Reducing Eventual Non-Termination). For a propositional formula \( \xi \) over the atoms \( \{p \triangleright 0 | p \in \mathbb{NPE}_S[\vec{x}], \triangleright \in \{\geq, >\}\} \), validity of

\[
\exists \vec{x} \in S^d, \ n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{n_0}. \ \xi
\]

(9)

can be reduced to validity of the closed formula \( \exists \vec{x} \in S^d. \ \text{red}(\xi) \in \text{Th}_3(S) \). This reduction takes polynomially many steps in the size of \( \xi \).

The time needed to compute the formula (9) is polynomial in the sum of the sizes of all poly-exponential expressions in \( \xi \). So the function (1) is computable in polynomial time w.r.t. the size of its input: \( \vec{q}_{\text{npe}} \) can clearly be computed in polynomial time from \( \vec{q} \) and we can then apply Thm. 4.9 to \( \varphi(\vec{q}_{\text{npe}}) \). Combining Cor. 3.3, (4), and Thm. 4.9 leads to the main result of this section.

**Theorem 4.10** (Reducing Termination). Termination of \( \text{npe}-\)loops (resp. \( \text{twn}-\)loops) on \( S^d \) is reducible to \( \text{Th}_3(S) \).
However, if the update contains *non-linear* terms, then its closed form and hence this reduction are not always computable in polynomial space (and thus, also not in polynomial time). Consider the following *twn*-loop $L_{\text{non-pspace}}$:

\[
\text{while (true) do } (x_1, x_2, \ldots, x_{d-1}, x_d) \leftarrow (x_d^d, x_d^3, \ldots, x_d, x_d^d, d \cdot x_d)
\]

The closed form for $x_i$ (i.e., the value of $x_i$ after $n$ loop iterations) is $q_i = d^{(d-1)(n-d+i)} \cdot x_d^{d-1}$ for all $n \geq d$. Thus, the closed form $q_1$ for $x_1$ contains constants like $d^{(d-1)}$ whose logarithm grows faster than any polynomial in $d$. Hence, $q_1$ cannot be computed in polynomial space.

Instead of computing closed forms directly, one could first linearize the loop (see [47] and Sect. 6) and then compute the closed form for the resulting linear-update loop. However, this approach cannot be computed in polynomial space either, because the linearization of $L_{\text{non-pspace}}$ contains the constant $d^{(d-1)}$ as well (see Ex. 6.14 in Sect. 6). We refer to Sect. 7 for an analysis of the complexity of deciding termination for *twn*-loops.

Our technique does not yield witnesses for non-termination, but the formula constructed by Thm. 4.9 describes the set of all witnesses for *eventual* non-termination. This set can be characterized by a formula from $\text{Th}_3(S)$ (i.e., it is $\text{Th}_3(S)$-definable, see Sect. 5.1), while in general the set of witnesses for non-termination cannot be characterized in this way (see [14]).

\[
\text{Lemma 4.12. Let } \xi = \varphi(\vec{q}_\text{norm}). \text{ Then } \vec{v} \in S^d \text{ witnesses eventual non-termination of } (\varphi, \vec{u}) \text{ on } S^d \text{ iff } \text{red}(\xi)(\vec{v}) \text{ holds.}
\]

In [28], we show how to compute witnesses for non-termination from witnesses for eventual non-termination of *twn*-loops. Thus, combining Lemma 4.12 with [28] shows that $\text{NT}_{(\varphi, \vec{u})}$ is recursively enumerable for *twn*-loops over $Z \leq S \leq R_A$. Alg. 1 summarizes our technique to check termination of *twn*-loops.

### 5 Transformation to Triangular Weakly Non-Linear Form

In this section, we show how to handle loops that are not yet *twn*. To this end, we introduce a transformation of loops via *polynomial automorphisms*
in Sect. 5.1 and show that our transformation preserves (non-)termination (Thm. 5.10). In Sect. 5.2, we use results from algebraic geometry to show that the question whether a loop can be transformed into *twm*-form is reducible to validity of Thm(\(\mathbb{R}\_k\))-formulas (Thm. 5.19). Moreover, we show that it is decidable whether a *linear* automorphism can transform a loop into a special case of the *twm*-form (Thm. 5.22). Finally, based on the transformation of Sect. 5.1 and 5.2 we generalize our results from Sect. 4 to certain non-*twm* loops in Sect. 5.3.

5.1 Transforming Loops

Clearly, the polynomials \(x_1, \ldots, x_d\) are *generators* of the \(S\)-algebra \(S[\vec{x}]\), i.e., every polynomial from \(S[\vec{x}]\) can be obtained from \(x_1, \ldots, x_d\) and the operations of the algebra (i.e., addition and multiplication). So far, we have implicitly chosen a special “representation” of the loop based on the generators \(x_1, \ldots, x_d\).

We now change this representation, i.e., we use \(d\) different polynomials which are also generators of \(S[\vec{x}]\). Then the loop has to be modified accordingly to adapt it to this new representation. This modification does not affect the loop’s termination behavior, but it may transform a non-*twm*-loop into *twm*-form. This change of representation is described by *\(S\)-automorphisms* of \(S[\vec{x}]\).

**Definition 5.1 (**\(S\)-Endomorphisms**). A mapping \(\eta: S[\vec{x}] \to S[\vec{x}]\) is an *\(S\*)-endomorphism* of \(S[\vec{x}]\) if it is *\(S\*)-linear* and *\(S\*)-multiplicative*, i.e., for all \(c, c' \in S\) and all \(p, p' \in S[\vec{x}]\) we have \(\eta(c \cdot p + c' \cdot p') = c \cdot \eta(p) + c' \cdot \eta(p')\), \(\eta(1) = 1\), and \(\eta(p \cdot p') = \eta(p) \cdot \eta(p')\). We denote the set of all *\(S\*)-endomorphisms* of \(S[\vec{x}]\) by \(\text{End}_S(S[\vec{x}])\). The set \(\text{Aut}_S(S[\vec{x}])\) of *\(S\*)-automorphisms* of \(S[\vec{x}]\) consists of those \(\eta \in \text{End}_S(S[\vec{x}])\) which are *invertible*, i.e., there exists an \(\eta^{-1} \in \text{End}_S(S[\vec{x}])\) with \(\eta \circ \eta^{-1} = \eta^{-1} \circ \eta = \text{id}_{S[\vec{x}]}\), where \(\text{id}_{S[\vec{x}]}\) is the identity function on \(S[\vec{x}]\).

**Example 5.2.** Let \(\eta \in \text{End}_S(S[x_1, x_2])\) with \(\eta(x_1) = x_2\) and \(\eta(x_2) = x_1 - x_2^2\). Then \(\eta \in \text{Aut}_S(S[x_1, x_2])\), whereas \(\eta^{-1}(x_1) = x_1^2 + x_2\) and \(\eta^{-1}(x_2) = x_1\).

As \(S[\vec{x}]\) is free on the generators \(\vec{x}\), an endomorphism \(\eta \in \text{End}_S(S[\vec{x}])\) is uniquely determined by the images of the variables, i.e., by \(\eta(x_1), \ldots, \eta(x_d)\). Hence, we have a one-to-one correspondence between elements of \((S[\vec{x}])^d\) and \(\text{End}_S(S[\vec{x}])\). In particular, every tuple \(\vec{u} = (u_1, \ldots, u_d) \in (S[\vec{x}])^d\) corresponds

---

**Algorithm 1:** Checking Termination

**Input:** a *twm*-loop \((\phi, \vec{u})\)

**Result:** \(\top\) resp. \(\perp\) if (non-)termination of \((\phi, \vec{u})\) on \(S^d\) is proven, \(?\) otherwise

if \((\phi, \vec{u})\) is not *tnn* then \((\phi, \vec{u}) \leftarrow (\phi \land \phi'(\vec{u}), \vec{u}(\vec{u}))\)

\(\vec{q} \leftarrow \text{closed form for } \vec{u}\)

if (in)validity of \(\exists \vec{x} \in S^d. \text{red}(\phi(\vec{q}_\text{norm}))\) cannot be proven then return ?

if \(\exists \vec{x} \in S^d. \text{red}(\phi(\vec{q}_\text{norm}))\) is valid then return \(\top\) else return \(\perp\)**
to the unique endomorphism \( \tilde{u} \in \text{End}_S(S[\tilde{x}]) \) which is defined as follows:

\[
\tilde{u}(x_i) = u_i \quad \text{for all} \ 1 \leq i \leq d
\]

So for any \( p \in S[\tilde{x}] \) we have \( \tilde{u}(p) = p(\tilde{u}) \). Thus, the update of a loop induces an endomorphism which operates on polynomials.

**Example 5.3.** Consider the loop \( L_{\text{out}} = (\varphi, \tilde{u}) \) where \( \tilde{u} = (u_1, u_2) \):

\[
\text{while} \ (x_2^3 + x_1 - x_2^2 > 0) \ \text{do} \ [x_2] \leftarrow \left[ \begin{array}{c}
\left( -x_2^3 + x_1 \right)^2 + x_2^3 - 2 \cdot x_2^2 + 2 \cdot x_1 \\
- \left( -x_2^3 + x_1 \right)^2 + x_2
\end{array} \right] \quad (11)
\]

Then \( \tilde{u} \) induces the endomorphism \( \tilde{u'} \) with \( \tilde{u'}(x_1) = u_1 \) and \( \tilde{u'}(x_2) = u_2 \). So we have \( \tilde{u'}(2 \cdot x_1 + x_2^3) = (2 \cdot u_1 + u_2^2) \).

Therefore, for a tuple of numbers like \( \tilde{c} = (5, 2) \), the induced endomorphism \( \tilde{c} \) is \( \tilde{c}(x_1) = 5 \) and \( \tilde{c}(x_2) = 2 \). Thus, we have \( \tilde{c}(x_2^3 + x_1 - x_2^2) = (5^3 + 2 - 2^2) = 2^3 + 5 - 2^2 = 9 \).

We extend the application of endomorphisms \( \eta : S[\tilde{x}] \to S[\tilde{x}] \) to vectors of polynomials \( \tilde{u} = (u_1, \ldots, u_d) \) by defining \( \eta(\tilde{u}) = (\eta(u_1), \ldots, \eta(u_d)) \) and to formulas \( \varphi \in \text{Th}_3(S, \mathbb{R}_p) \) by defining \( \eta(\varphi) = \varphi(\eta(\tilde{x})) \), i.e., \( \eta(\varphi) \) results from \( \varphi \) by applying \( \eta \) to all polynomials that occur in \( \varphi \). This allows us to transform \( (\varphi, \tilde{u}) \) into a new loop \( Tr_\eta(\varphi, \tilde{u}) \) using any automorphism \( \eta \in \text{Aut}_S(S[\tilde{x}]) \).

**Definition 5.4 (Tr).** For \( \eta \in \text{Aut}_S(S[\tilde{x}]) \), we define \( Tr_\eta(\varphi, \tilde{u}) = (\varphi', \tilde{u}') \) where

\[
\varphi' = \eta^{-1}(\varphi) \quad \text{and} \quad \tilde{u}' = (\eta^{-1} \circ \tilde{u} \circ \eta)(\tilde{x}).
\]

In other words, we have \( \tilde{u'} = (\eta(\tilde{x}))(\tilde{u})(\eta^{-1}(\tilde{x})) \) since \( (\eta^{-1} \circ \tilde{u} \circ \eta)(\tilde{x}) = \eta^{-1}(\eta(\tilde{x})(\tilde{u})(\eta^{-1}(\tilde{x}))) = \eta^{-1}(\eta(\tilde{x})) \circ \eta^{-1}(\tilde{x}) \).

**Example 5.5.** We transform the loop \( L_{\text{out}} \) in (11) with the automorphism \( \eta \) from Ex. 5.2. We obtain \( Tr_\eta(\varphi, \tilde{u}) = (\varphi', \tilde{u}') \) where

\[
\varphi' = \eta^{-1}(\varphi) = (\eta^{-1}(x_2^3))^3 + \eta^{-1}(x_1) - (\eta^{-1}(x_2))^3 > 0 = (x_1^3 + x_2^2 - x_1 > 0) \quad \text{and}
\]

\[
\tilde{u}' = ((\eta^{-1} \circ \tilde{u} \circ \eta)(x_1), (\eta^{-1} \circ \tilde{u} \circ \eta)(x_2)) = (\eta^{-1}(\tilde{u}(x_1)), \eta^{-1}(\tilde{u}(x_2))) = \eta^{-1}(\tilde{u}(x_1 - u_2^2)) = (x_1 + x_2^2, 2 \cdot x_2).
\]

So the resulting transformed loop is \( (x_1^3 + x_2 > 0, x_1 + x_2^2, 2 \cdot x_2) \). Note that while the original loop \( (\varphi, \tilde{u}) \) is neither triangular nor weakly non-linear, the resulting transformed loop is tun. Also note that we used a non-linear automorphism with \( \eta(x_2) = x_1 - x_2^2 \) for the transformation.

While the above example shows that our transformation can indeed transform non-tun-loops into tun-loops, it remains to prove that this transformation preserves (non-)termination. Then we can use our techniques for termination analysis of tun-loops for tun-transformable-loops as well, i.e., for all loops \( (\varphi, \tilde{u}) \) where \( Tr_\eta(\varphi, \tilde{u}) \) is tun for some automorphism \( \eta \). (The question how to find such automorphisms will be addressed in Sect. 5.2.)

As a first step, by Lemma 5.6, our transformation is “compatible” with the operation \( \circ \) of the group \( \text{Aut}_S(S[\tilde{x}]) \), i.e., it is an action.
Lemma 5.6. $\text{Aut}_S(S[\vec{x}])$ acts via $\text{Tr}$ on loops, i.e., for $\eta_1, \eta_2 \in \text{Aut}_S(S[\vec{x}])$, we have $\text{Tr}_{\text{id} \circ \eta_1}(\varphi, \vec{u}) = (\varphi, \vec{u})$ and $\text{Tr}_{\eta_1 \circ \eta_2}(\varphi, \vec{u}) = \text{Tr}_{\eta_1}(\text{Tr}_{\eta_2}(\varphi, \vec{u}))$.

The following lemma will enable us to generalize our results on witnesses for (eventual) non-termination to loops which can be transformed into $\text{twm}$-form.

Lemma 5.7. Let $\text{Tr}_{\eta}(\varphi, \vec{u}) = (\varphi', \vec{u}')$ and let $\hat{\eta} : S^d \to S^d$ map $\vec{c}$ to $\hat{\eta}(\vec{c}) = \hat{\vec{c}}(\eta(\vec{x})) = \eta(\vec{x}) \hat{\vec{c}}$. Then $\varphi(\hat{\vec{c}}(\vec{c})) = \varphi'((\vec{u}')^\eta(\hat{\vec{c}}))$ for all $\vec{c} \in S^d$ and $n \in \mathbb{N}$.

Lemma 5.7 yields the following corollary which shows that $\eta(\vec{x})$ transforms witnesses for (eventual) non-termination of $(\varphi, \vec{u})$ into witnesses for $\text{Tr}_{\eta}(\varphi, \vec{u})$.

Corollary 5.8. If $\vec{c}$ witnesses (eventual) non-termination of $(\varphi, \vec{u})$, then $\hat{\eta}(\vec{c})$ witnesses (eventual) non-termination of $\text{Tr}_{\eta}(\varphi, \vec{u})$.

Example 5.9. For the tuple $\vec{c} = (5, 2)$ from Ex. 5.3 and the automorphism $\eta$ from Ex. 5.2 with $\eta(x_1) = x_2$ and $\eta(x_2) = x_1 - x_2^2$, we obtain

\[ \hat{\eta}(\vec{c}) = (\eta(x_1), \eta(x_2)) (\vec{c}) = (2, 5 - 2^2) = (2, 1). \]

As $\vec{c} = (5, 2)$ witnesses non-termination of the loop $L_{\text{aut}} = (\varphi, \vec{u})$ in (11), $\hat{\eta}(\vec{c}) = (2, 1)$ witnesses nontermination of $\text{Tr}_{\eta}(\varphi, \vec{u})$ due to Cor. 5.8.

Finally, Thm. 5.10 states that transforming loops preserves (non-)termination.

Theorem 5.10 (Tr Preserves Termination). If $\eta \in \text{Aut}_S(S[\vec{x}])$, then $\hat{\eta}$ is a bijection between the respective sets of witnesses for (eventual) non-termination, i.e., for $\text{Tr}_{\eta}(\varphi, \vec{u}) = (\varphi', \vec{u}')$ we have $\vec{c} \in (E)\text{NT}_{\varphi, \vec{u}}$ iff $\hat{\eta}(\vec{c}) \in (E)\text{NT}_{\varphi', \vec{u}'}$. Therefore, $(\varphi, \vec{u})$ terminates iff $\text{Tr}_{\eta}(\varphi, \vec{u}) = (\varphi', \vec{u}')$ terminates.

Up to now, we only transformed a loop $(\varphi, \vec{u})$ on $S^d$ using elements of $\text{Aut}_S(S[\vec{x}])$. To see that this is a limitation, consider a linear-update loop where $\vec{u} = A \cdot \vec{x}$ and $A$ only has real eigenvalues. In Sect. 7.1 we will show that these loops can always be transformed into $\text{twm}$-form and a suitable automorphism $\eta$ can be obtained by computing the Jordan normal form of $A$. This automorphism $\eta$ is only an element of $\text{Aut}_S(S[\vec{x}])$ if the eigenvalues of $A$ are from $S$. So if $S = \mathbb{Z}$, then this transformation is only applicable if all eigenvalues of $A$ are integers.

However, we can also transform $(\varphi, \vec{u})$ into the loop $\text{Tr}_{\eta}(\varphi, \vec{u})$ on $\mathbb{R}_d^d$ using an automorphism $\eta \in \text{Aut}_{\mathbb{R}_d^d}(\mathbb{R}_d[\vec{x}])$. Nevertheless, our goal remains to prove termination on $S^d$ instead of $\mathbb{R}_d^d$, which is not equivalent in general. Thus, in Sect. 5.3 we will show how to analyze termination of loops on certain subsets $F$ of $\mathbb{R}_d^d$. This allows us to analyze termination of $(\varphi, \vec{u})$ on $S^d$ by checking termination of $\text{Tr}_{\eta}(\varphi, \vec{u})$ on the subset $\hat{\eta}(S^d) \subseteq \mathbb{R}_d^d$ instead.

By our definition of loops over a ring $S$, we have $\vec{u}(\vec{c}) \in S^d$ for all $\vec{c} \in S^d$, i.e., $S^d$ is $\vec{u}$-invariant. This property is preserved by our transformation.

Definition 5.11 ($\vec{u}$-Invariance). Let $(\varphi, \vec{u})$ be a loop on $S^d$ and let $F \subseteq S^d$. We call $F$ $\vec{u}$-invariant or update-invariant if for all $\vec{c} \in F$ we have $\vec{u}(\vec{c}) \in F$. 

Lemma 5.12. Let $(\varphi, \vec{u})$ be a loop on $S^d$, $F \subseteq S^d$ be $\vec{u}$-invariant, $\eta \in \text{Aut}_{R_\mathbb{A}}(\mathbb{A}[\vec{x}])$, and let $\text{Tr}_\eta(\varphi, \vec{u}) = (\varphi', \vec{u}')$. Then $\hat{\eta}(F)$ is $\vec{u}'$-invariant.

Our goal is to reduce termination to a $\text{Th}_3(S, \mathbb{A})$-formula. Clearly, termination on $F$ cannot be encoded with such a formula if $F$ cannot be defined via $\text{Th}_3(S, \mathbb{A})$. Thus, we require that $F$ is $\text{Th}_3(S, \mathbb{A})$-definable, i.e., that there is a $\psi \in \text{Th}_3(S, \mathbb{A})$ with free variables $\vec{x}$ such that we have $c \in F$ iff $\psi(c)$ is valid. Then we also say that $\psi$ defines $F$. An example for a $\text{Th}_3(Z, \mathbb{A})$-definable set is $\{(a, 0) \mid a \in \mathbb{Z}\}$, which is characterized by the formula $\exists a \in \mathbb{Z}. x_1 = a \land x_2 = 0 \land x_3 = a$.

To analyze termination of $(\varphi, \vec{u})$ on $S^d$, we can analyze termination of $\text{Tr}_\eta(\varphi, \vec{u})$ on $\hat{\eta}(S^d) \subseteq \mathbb{A}^d$ instead. The reason is that $c \in S^d$ is a witness for (eventual) non-termination of $(\varphi, \vec{u})$ iff $\hat{\eta}(c)$ is a witness for $\text{Tr}_\eta(\varphi, \vec{u})$ due to Cor. 5.8, i.e., $S^d$ contains a witness for (eventual) non-termination of $(\varphi, \vec{u})$ iff $\hat{\eta}(S^d)$ contains a witness for $\text{Tr}_\eta(\varphi, \vec{u})$. While $S^d$ is clearly $\text{Th}_3(S, \mathbb{A})$-definable, Lemma 5.13 shows that $\hat{\eta}(S^d)$ is $\text{Th}_3(S, \mathbb{A})$-definable, too. More precisely, $\text{Th}_3(S, \mathbb{A})$-definability is preserved by polynomial endomorphisms.

Lemma 5.13. Let $\mathbb{Z} \leq S \leq \mathbb{A}$ and let $\eta \in \text{End}_{\mathbb{A}}(\mathbb{A}[\vec{x}])$. If $F \subseteq \mathbb{A}^d$ is $\text{Th}_3(S, \mathbb{A})$-definable then so is $\hat{\eta}(F)$.

Example 5.14. The set $\mathbb{Z}^2$ is $\text{Th}_3(Z, \mathbb{A})$-definable as we have $(x_1, x_2) \in \mathbb{Z}^2$ iff $\exists a, b \in \mathbb{Z}. x_1 = a \land x_2 = b$.

Let $\eta \in \text{End}_{\mathbb{A}}(\mathbb{A}[\vec{x}])$ with $\eta(x_1) = \frac{1}{2} \cdot x_1^2 + x_2^2$ and $\eta(x_2) = x_3^2$. Then $\hat{\eta}(\mathbb{Z}^2)$ is also $\text{Th}_3(Z, \mathbb{A})$-definable because for $x_1, x_2 \in \mathbb{A}$ we have $(x_1, x_2) \in \eta(\mathbb{Z}^2)$ iff $\exists y_1, y_2 \in \mathbb{A}. a, b \in \mathbb{Z}. y_1 = a \land y_2 = b \land x_1 = \frac{1}{2} \cdot y_1^2 + y_2^2 \land x_2 = y_2^2$.

Thm. 5.15 shows that instead of regarding solvable loops [55], w.l.o.g. we can restrict ourselves to twn-loops. The reason is that every solvable loop with real eigenvalues can be transformed into a twn-loop by a linear automorphism $\eta$, i.e., the degree $\deg(\eta)$ of $\eta$ is 1, where $\deg(\eta) = \max(\deg(\eta(x_i)) \mid 1 \leq i \leq d)$. Hence, the twn-loops can be transformed into solvable loops.

Theorem 5.15 (twn-Transformability of Solvable Loops). Let $(\varphi, \vec{u})$ be a solvable loop with real eigenvalues. Then one can compute a linear automorphism $\eta \in \text{Aut}_{\mathbb{A}}(\mathbb{A}[\vec{x}])$ such that $\text{Tr}_\eta(\varphi, \vec{u})$ is twn.

We recapitulate our most important results on $\text{Tr}$ in the following corollary. Here, we generalize the result of Thm. 5.10 to the setting where we consider termination on some update-invariant and $\text{Th}_3(S, \mathbb{A})$-definable set.

Corollary 5.16 (Properties of $\text{Tr}$). Let $(\varphi, \vec{u})$ be a loop, $\eta \in \text{Aut}_{\mathbb{A}}(\mathbb{A}[\vec{x}])$, $\text{Tr}_\eta(\varphi, \vec{u}) = (\varphi', \vec{u}')$, and $F \subseteq S^d$ be $\vec{u}$-invariant and $\text{Th}_3(S, \mathbb{A})$-definable.

(a) $\hat{\eta}(F) \subseteq \mathbb{A}^d$ is $\vec{u}'$-invariant and $\text{Th}_3(S, \mathbb{A})$-definable.
(b) $(\varphi, \vec{u})$ terminates on $F$ iff $(\varphi', \vec{u}')$ terminates on $\hat{\eta}(F)$.
(c) $\hat{\eta}(\vec{c}) \in \hat{\eta}(F)$ witnesses (eventual) non-termination of $(\varphi, \vec{u})$ iff $\hat{\eta}(\vec{c}) \in \hat{\eta}(F)$ witnesses (eventual) non-termination of $(\varphi', \vec{u}')$. 

5.2 Finding Automorphisms to Transform Loops into \textit{twn}-Form

The goal of \textit{Tr} from Sect. 5.1 is to transform \((\varphi, \vec{u})\) into \textit{twn}-form such that our techniques from Sect. 4 can be used to decide termination. So the two remaining challenges are to find a suitable automorphism \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\) such that \(\text{Tr}_\eta (\varphi, \vec{u})\) is \textit{twn}, and to adapt our techniques from Sect. 4 such that they can be applied to \textit{twn}-loops where one only wants to show termination on an update-invariant \(\text{Th}_{\exists} (\mathcal{S}, \mathbb{R}_{\mathcal{A}})\)-definable subset. We discuss the latter challenge in Sect. 5.3. In this section, we present two techniques to check the existence of automorphisms for the transformation into \textit{twn}-form constructively, i.e., these techniques can also be used to compute such automorphisms.

The search for suitable automorphisms is closely related to the question if a polynomial automorphism can be conjugated into a “de Jonguières”-automorphism, a difficult question from algebraic geometry \[18\]. So future advances in this field may help to improve the results of the current section.

The first technique (Thm. 5.19) reduces the search for a suitable automorphism of bounded degree to \(\text{Th}_{\exists} (\mathbb{R}_{\mathcal{A}})\). For any automorphism, the degree of its inverse is bounded in terms of the length \(d\) of \(\vec{x}\).

\textbf{Theorem 5.17 (Degree of Inverse \[18, \text{Cor. 2.3.4}]\).} Let \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\). Then we have \(\deg(\eta^{-1}) \leq (\deg(\eta))^{d-1}\).

By Thm. 5.17, checking if an endomorphism is an automorphism can be reduced to \(\text{Th}_{\exists} (\mathbb{R}_{\mathcal{A}})\). To do so, one encodes the existence of coefficients for the polynomials \(\eta^{-1}(x_1), \ldots, \eta^{-1}(x_d)\), which all have at most degree \((\deg(\eta))^{d-1}\).

\textbf{Lemma 5.18.} Let \(\eta \in \text{End}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\). Then the question if \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\) holds is reducible to \(\text{Th}_{\exists} (\mathbb{R}_{\mathcal{A}})\).

Based on Lemma 5.18, we now present our first technique to find an automorphism \(\eta\) that transforms a loop into \textit{twn}-form.

\textbf{Theorem 5.19 (Tr With Automorphisms of Bounded Degree).} For any \(\delta \geq 0\), the question whether there exists an \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\) with \(\deg(\eta) \leq \delta\) such that \(\text{Tr}_\eta (\varphi, \vec{u})\) is \textit{twn} is reducible to \(\text{Th}_{\exists} (\mathbb{R}_{\mathcal{A}})\).

So if the degree of \(\eta\) is bounded a priori, it is decidable whether there exists an \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\) such that \(\text{Tr}_\eta (\varphi, \vec{u})\) is \textit{twn} since \(\text{Th}_{\exists} (\mathbb{R}_{\mathcal{A}})\) is decidable.

We call a loop \textit{twn-transformable} if there is an \(\eta \in \text{Aut}_{\mathbb{R}_{\mathcal{A}}} (\mathbb{R}_{\mathcal{A}}[\vec{x}])\) such that \(\text{Tr}_\eta (\varphi, \vec{u})\) is \textit{twn}. By Thm. 5.19, \textit{twn}-transformability is \textit{semi}-decidable since one can increment \(\delta\) until a suitable automorphism is found. So in other words, any loop which is transformable to a \textit{twn}-loop can be transformed via Thm. 5.19.

We call our transformation \textit{Tr complete} for a class of loops if every loop from this class is \textit{twn}-transformable. For such classes, a suitable automorphism is computable by Thm. 5.19. Together with Thm. 5.15, we get Cor. 5.20.

\textbf{Corollary 5.20.} \textit{Tr} is complete for solvable loops with real eigenvalues.
while \((4 \cdot x_2^2 + x_1 + x_2 + x_3 > 0)\) do \((x_1, x_2, x_3) \leftarrow (u_1, u_2, u_3)\)

with \(u_1 = x_1 + 8 \cdot x_1 \cdot x_2^2 + 16 \cdot x_2^3 + 16 \cdot x_2^2 \cdot x_3\)
\(u_2 = x_2 - x_1^2 - 4 \cdot x_1 \cdot x_2 - 4 \cdot x_1 \cdot x_3 - 4 \cdot x_2^2 - 8 \cdot x_2 \cdot x_3 - 4 \cdot x_3^2\)
\(u_3 = x_3 - 4 \cdot x_1 \cdot x_2^2 - 8 \cdot x_3^3 - 8 \cdot x_2^2 \cdot x_3 + x_1^2 + 4 \cdot x_1 \cdot x_2 + 4 \cdot x_1 \cdot x_3 + 4 \cdot x_3^2\)

Fig. 3: Loop \(L_{\text{non-twn}}\)

Note that for solvable loops \((\varphi, \vec{u})\), instead of computing \(\eta\) using Thm. 5.19, the proof of Thm. 5.15 yields a more efficient way to compute a linear automorphism \(\eta\) such that \(\text{Tr}_\eta(\varphi, \vec{u})\) is twn. For this, one computes the Jordan normal form of each \(A_i\) (see Def. 2.3), which is possible in polynomial time (see, e.g., [23, 53]).

Our second technique to find suitable automorphisms for our transformation is restricted to linear automorphisms. In this case, it is decidable whether a loop can be transformed into a twn-loop \((\varphi', \vec{u}')\) where the monomial for \(x_i\) has the coefficient 1 in each \(u_i'\). The decision procedure checks if a certain Jacobian matrix is strongly nilpotent, i.e., it is not based on a reduction to \(\text{Th}_3(\mathbb{R}_\lambda)\).

**Definition 5.21 (Strong Nilpotence).** Let \(J \in (\mathbb{R}_\lambda[\vec{x}])^{d \times d}\) be a matrix of polynomials. For all \(1 \leq i \leq d\), let \(\vec{y}^{(i)}\) be a vector of fresh variables. \(J\) is strongly nilpotent if \(\prod_{i=1}^d J[\vec{x}/\vec{y}^{(i)}] = 0^{d \times d}\), where \(0^{d \times d}\) is the zero matrix.

Our second technique is formulated in the following theorem which follows from an existing result in linear algebra [17, Thm. 1.6].

**Theorem 5.22 (Tr With Linear Automorphisms [17]).** Let \((\varphi, \vec{u})\) be a loop. The Jacobian matrix \(\left(\frac{\partial(u_i-x_1)}{\partial x_i}\right)_{1 \leq i, j \leq d} \in (\mathbb{R}_\lambda[\vec{x}])^{d \times d}\) is strongly nilpotent iff there exists a linear automorphism \(\eta \in \text{Aut}_{\lambda}(\mathbb{R}_\lambda[\vec{x}])\) with
\[
\text{Tr}_\eta(\varphi, \vec{u}) = (\varphi', (x_1 + p_1, \ldots, x_d + p_d))
\]
and \(p_i \in \mathbb{R}_\lambda[x_{i+1}, \ldots, x_d]\) for all \(1 \leq i \leq d\). Thus, \(\text{Tr}_\eta(\varphi, \vec{u})\) is twn.

As strong nilpotence of the Jacobian matrix is clearly decidable, Thm. 5.22 gives rise to a decision procedure for the existence of a linear automorphism that transforms \((\varphi, \vec{u})\) to the form (12).

**Example 5.23.** The loop \(L_{\text{non-twn}}\) on \(S^3\) in Fig. 3 shows how our results enlarge the class of loops where termination is reducible to \(\text{Th}_3(S, \mathbb{R}_\lambda)\). This loop is clearly not in twn-form. To transform it, we use Thm. 5.22. The Jacobian matrix \(J\) of \((u_1 = x_1, u_2 = x_2, u_3 = x_3)\) is:
\[
\begin{bmatrix}
8 \cdot x_2^3 & 16 \cdot x_1 + 48 \cdot x_2^2 + 32 \cdot x_3 \cdot x_3 & 16 \cdot x_2^2 \\
-2 \cdot x_1 - 4 \cdot x_2 - 4 \cdot x_3 & -4 \cdot x_1 - 8 \cdot x_2 - 8 \cdot x_3 & -4 \cdot x_1 - 8 \cdot x_2 - 8 \cdot x_3 \\
-4 \cdot x_2^2 + 2 \cdot x_1 + 4 \cdot x_2 + 4 \cdot x_3 & -8 \cdot x_1 - 24 \cdot x_2^2 - 16 \cdot x_2 + 4 \cdot x_1 + 8 \cdot x_2 + 8 \cdot x_3 & -8 \cdot x_2^2 + 4 \cdot x_1 + 8 \cdot x_2 + 8 \cdot x_3
\end{bmatrix}
\]

One easily checks that \(J\) is strongly nilpotent. Thus, by Thm. 5.22 the loop can be transformed into twn-form by a linear automorphism. Indeed, consider
the linear automorphism \( \eta \in \text{Aut}_{\mathbb{R}_A}(\mathbb{R}_A[\mathbf{x}]) \) induced by the matrix \( M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 2 & 2 \end{bmatrix} \), i.e.,

\[
\begin{align*}
\eta &: \quad x_1 \mapsto x_1 + x_2 + x_3, \quad x_2 \mapsto 2 \cdot x_2, \quad x_3 \mapsto x_1 + 2 \cdot x_2 + 2 \cdot x_3 \\
\eta^{-1} &: \quad x_1 \mapsto 2 \cdot x_1 - x_3, \quad x_2 \mapsto \frac{1}{2} \cdot x_2, \quad x_3 \mapsto -x_1 - \frac{1}{2} \cdot x_2 + x_3.
\end{align*}
\]

If we transform \( L_{\text{non-twn}} \) with \( \eta \), we obtain the twn-loop \( \mathbf{L}_{\text{ex}} \) in Fig. 2. If \( S = \mathbb{R}_A \), then \( \mathbf{L}_{\text{ex}} \) terminates on \( \mathbb{R}_A^3 \) iff \( L_{\text{non-twn}} \) terminates on \( \mathbb{R}_A^3 \) by Thm. 5.10. Thus, as seen in Ex. 4.8, \( L_{\text{non-twn}} \) does not terminate on \( \mathbb{R}_A^3 \). Now assume \( S = \mathbb{Z} \), i.e., we analyze termination of \( L_{\text{non-twn}} \) on \( \mathbb{Z}^3 \) instead of \( \mathbb{R}_A^3 \). Note that \( \hat{\eta} \) maps \( \mathbb{Z}^3 \) to the set of all \( \mathbb{Z} \)-linear combinations of columns of \( M \), i.e.,

\[
\hat{\eta}(\mathbb{Z}^3) = \{ a \cdot (1, 0, 1) + b \cdot (1, 2, 2) + c \cdot (1, 0, 2) \mid a, b, c \in \mathbb{Z} \}.
\]

By Cor. 5.16, \( L_{\text{ex}} \) terminates on \( \hat{\eta}(\mathbb{Z}^3) \) iff \( L_{\text{non-twn}} \) terminates on \( \mathbb{Z}^3 \). Moreover, \( \hat{\eta}(\mathbb{Z}^3) \) is \( \text{Th}_3(\mathbb{Z}, \mathbb{R}_A) \)-definable: We have \( (x_1, x_2, x_3) \in \hat{\eta}(\mathbb{Z}^3) \) iff

\[
\exists a, b, c \in \mathbb{Z}. \quad x_1 = a \cdot 1 + b \cdot 1 + c \cdot 1 \wedge x_2 = b \cdot 2 \wedge x_3 = a \cdot 1 + b \cdot 2 + c \cdot 2.
\]

We will discuss how to analyze termination of twn-loops like \( \mathbf{L}_{\text{ex}} \) on update-invariant and \( \text{Th}_3(\mathbb{Z}, \mathbb{R}_A) \)-definable sets like \( \hat{\eta}(\mathbb{Z}^3) \) in Sect. 5.3.

To summarize, if a loop is twn-transformable, then we can always find a suitable automorphism via Thm. 5.19. So whenever Thm. 5.22 is applicable, a suitable linear automorphism can also be obtained by Thm. 5.19. Hence, our first technique from Thm. 5.19 subsumes our second one from Thm. 5.22. However, while Thm. 5.19 is always applicable, Thm. 5.22 is easier to apply. The reason is that for Thm. 5.19 one has to check validity of a possibly non-linear formula over the reals, where the degree of the occurring polynomials depends on \( d \) and the update \( \mathbf{u} \) of the loop, and the number of variables can be exponential in \( d \), see Thm. 5.17 and Lemma 5.18. So even when searching for a linear automorphism, one may obtain a non-linear formula if the loop is non-linear. In contrast, Thm. 5.22 only requires linear algebra. So it is preferable to first check whether the loop can be transformed into a twn-loop \( (\varphi', (x_1 + p_1, \ldots, x_d + p_d)) \) with \( x_i \notin \mathcal{V}(p_i) \) via a linear automorphism. This check is decidable by Thm. 5.22.

Note that the proof of Thm. 5.19 is constructive. Moreover, the proof of [17, Thm. 1.6.] which implies Thm. 5.22 is also constructive: the idea is to use basic results from linear algebra to compute an invertible matrix \( T \in \mathbb{R}_A^{d \times d} \) such that \( T \cdot J \cdot T^{-1} \) is triangular where \( J \) is the Jacobian matrix \( \left( \frac{\partial u_i - u_i}{\partial x_j} \right)_{1 \leq i, j \leq d} \). Then \( \eta \) with \( \eta(\mathbf{x}) = T \cdot \mathbf{x} \) transforms the loop into the form (12). Hence, Thm. 5.22 is also constructive. Thus, we can not only check the existence of a suitable automorphism, but we can also compute it whenever it exists.
5.3 Analyzing \textit{twn}-Transformable Loops

In this section, we generalize our results from Sect. 4 to \textit{twn}-transformable loops. Our transformation from Sect. 5.1 and 5.2 transforms \textit{twn}-transformable loops over update-invariant and $\text{Th}_3(S, R_{\lambda})$-definable sets into \textit{twn}-loops over update-invariant and $\text{Th}_3(S, R_{\lambda})$-definable sets. Thus, in this section we fix a \textit{twn}-loop $(\varphi, \vec{u})$ and such a set $F \subseteq R_{\lambda}^d$. Let $\psi_F \in \text{Th}_3(S, R_{\lambda})$ define $F$.

While in Sect. 4 we were concerned with the termination of loops on a set $S^d$ for a ring $S$, we now show that termination of $(\varphi, \vec{u})$ on $F$ can also be reduced to an existential formula (from $\text{Th}_3(S, R_{\lambda})$). Here, we indeed rely on the update-invariance of $F$ as otherwise eventual non-termination and non-termination of $(\varphi, \vec{u})$ on $F$ would not be equivalent. In Sect. 4, this equivalence is crucial since we reduce non-termination via eventual non-termination to $\text{Th}_3(S, R_{\lambda})$.

Again, let $\vec{u}_{\text{norm}}$ be the normalized closed form of $\vec{u}$. Similar to (4), $(\varphi, \vec{u})$ is eventually non-terminating on $F$ iff

\begin{equation}
\exists \vec{z} \in F, n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. \varphi(\vec{u}_{\text{norm}}).
\end{equation}

In Thm. 4.9, we have seen that given a propositional formula $\xi$ over the atoms \{p \triangleright 0 \mid p \in \text{NPE}_2[\vec{z}], \triangleright \in \{\geq, >\}\}, one can reduce validity of $\exists \vec{z} \in S^d, n_0 \in \mathbb{N}, \forall n \in \mathbb{N}_{>n_0}, \xi$ to validity of $\exists \vec{z} \in S^d, \text{red}(\xi) \in \text{Th}_3(S)$ and the resulting formula can be computed in polynomial time from $\xi$. Thus, by using the formula $\exists \vec{z} \in R_{\lambda}^d, \psi_F \wedge \text{red}(\xi) \in \text{Th}_3(S, R_{\lambda})$ instead, we obtain Cor. 5.24.

\textbf{Corollary 5.24 (Reducing Eventual Non-Termination on a Set). } For a propositional formula $\xi$ over \{p \triangleright 0 \mid p \in \text{NPE}_2[\vec{z}], \triangleright \in \{\geq, >\}\}, validity of

\begin{equation}
\exists \vec{z} \in F, n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. \xi
\end{equation}

can be reduced to validity of a closed formula in $\text{Th}_3(S, R_{\lambda})$ in polynomial time.

Combining (14) and Cor. 5.24 yields the following refined version of Thm. 4.10.

\textbf{Corollary 5.25 (Reducing Termination on Sets). } Termination of \textit{tn}- or \textit{twn}-loops, respectively, on update-invariant $\text{Th}_3(S, R_{\lambda})$-definable sets is reducible to $\text{Th}_3(S, R_{\lambda})$.

\textbf{Example 5.26. } Consider Ex. 5.23, where we have seen that $L_{\text{non-\textit{twn}}}$ (see Fig. 3) terminates on $\mathbb{Z}^3$ iff $L_{\text{tn}}$ (see Fig. 2) terminates on the update-invariant and $\text{Th}_3(\mathbb{Z}, \mathbb{R})$-definable set $\tilde{u}(\mathbb{Z}^3) = F$ defined by the formula (13). In Ex. 4.8, we showed that $(-4, 2, 1)$ witnesses eventual non-termination of $L_{\text{tn}}$. As $\tilde{u}(-9, 1, 4) = (-9 + 1 + 4, 1 \cdot 2, -9 + 1 \cdot 2 + 4 \cdot 2) = (-4, 2, 1)$, we have $(-4, 2, 1) \in F$. Furthermore, $(-9, 1, 4)$ witnesses eventual non-termination of $L_{\text{non-\textit{twn}}}$ on $\mathbb{Z}^3$ by Cor. 5.16 (c). Hence, $L_{\text{non-\textit{twn}}}$ does not terminate on $\mathbb{Z}^3$.

In addition, we get the following refined version of Cor. 4.11.

\textbf{Corollary 5.27 (\textit{(Semi-)Deciding} (Non-)Termination on a Set). } Let $(\varphi, \vec{u})$ be a \textit{twn}-loop and let $F \subseteq R_{\lambda}^d$ be update-invariant and $\text{Th}_3(S, R_{\lambda})$-definable.
Lemma 4.12

(a) The loop $(\varphi, \vec{u})$ terminates over $\mathbb{R}_h$ iff it terminates over $\mathbb{R}$.
(b) Termination of $(\varphi, \vec{u})$ on $F$ is decidable if $S = \mathbb{R}_h$ or $S = \mathbb{R}$.
(c) Non-termination of $(\varphi, \vec{u})$ on $F$ is semi-decidable if $S = \mathbb{Z}$ or $S = \mathbb{Q}$.

Of course, Lemma 4.12 also holds in this setting.

Corollary 5.28. Let $\xi = \varphi(\vec{q}_{\text{norm}})$. Then $\vec{c} \in \mathbb{R}^d_+$ witnesses eventual non-termination of $(\varphi, \vec{u})$ on $F$ iff $\psi_F(\vec{c}) \wedge \text{red}(\xi)(\vec{c})$ holds.

Finally, Alg. 2 generalizes Alg. 1 to $\text{twn}$-transformable loops.

Algorithm 2: Checking Termination on Sets

| Input: | a $\text{twn}$-transformable-loop $(\varphi, \vec{u})$ and $\psi_F \in \text{Th}_2(S, \mathbb{R}_h)$ |
|--------|--------------------------------------------------|
| Result: | $\top$ resp. $\bot$ if (non-)termination of $(\varphi, \vec{u})$ on $F$ is proven, $\top$ otherwise |

$(\varphi, \vec{u}) \leftarrow \text{Tr}_n(\varphi, \vec{u})$, $\psi_F \leftarrow \psi_F[F]$, such that $(\varphi, \vec{u})$ becomes $\text{twn}$

if $(\varphi, \vec{u})$ is not $\text{twn}$ then $(\varphi, \vec{u}) \leftarrow (\varphi \land (\varphi(\vec{u})), \vec{u}(\vec{u}))$

$q \leftarrow$ closed form for $\vec{u}$

if (in)validity of $\exists \vec{x} \in \mathbb{R}^d_+, \psi_F \wedge \text{red}(\varphi(\vec{q}_{\text{norm}}))$ cannot be proven then return $\bot$

if $\exists \vec{x} \in \mathbb{R}^d_+, \psi_F \wedge \text{red}(\varphi(\vec{q}_{\text{norm}}))$ is valid then return $\bot$ else return $\top$

6 Linearization of $\text{twn}$-Loops

In [47], a technique was proposed to linearize polynomial loops. As mentioned, by combining the linearization with existing decidability results [63, 64], one can conclude decidability of termination for conjunctive $\text{twn}$-loops over $\mathbb{R}$ (whereas Cor. 4.11 extends this result also to non-conjunctive loops).

In this section, we adapt the linearization technique of [47] to our setting and formalize it. This allows us to obtain novel results on the complexity of linearization which we use to analyze the complexity of deciding termination for arbitrary $\text{twn}$-loops in Sect. 7.2. We start with the definition of linearization.

Definition 6.1 (Linearization). Let $\vec{u} \in (S[\vec{x}])^d$ and $\vec{g}$ be a vector of $d'$ fresh variables with $d' \geq d$. Let $\vec{u}' \in (S[\vec{y}])^d$ be linear and $\vec{w} \in (S[\vec{x}])^d$ with $w_i = x_i$ for all $1 \leq i \leq d$. Then $\vec{u}'$ is a linearization of $\vec{u}$ via $\vec{w}$ if $\vec{w}(\vec{u}(\vec{c})) = \vec{u}'(\vec{w}(\vec{c}))$ holds for all $\vec{c} \in S^d$, where $\vec{u}'(\vec{w}(\vec{c}))$ stands for $\vec{u}'[\vec{g}/\vec{w}(\vec{c})]$. Instead of $y_i$, we often write $w_i$, for all $1 \leq i \leq d'$.

So $y_1, \ldots, y_d$ (i.e., $w_1, \ldots, w_d$) correspond to the variables $x_1, \ldots, x_d$, whereas $y_{d+1}, \ldots, y_{d'}$ are used to mimic the non-linear part of $\vec{u}$ in a linear way in $\vec{w}$’. This non-linear behavior is captured by the polynomials $w_{d+1}, \ldots, w_{d'}$.

Example 6.2. Let $\vec{u} = (x_2^2, x_3^3, x_3) \in (\mathbb{Z}[x_1, x_2, x_3])^3$. Then $\vec{u}' = (y_{x_2}^2, y_{x_3}^3, y_{x_3}, y_{x_2}^2, y_{x_3})$ over the variables $(y_{x_1}, y_{x_2}, y_{x_3}, y_{x_2}^2, y_{x_3})$ is a linearization of $\vec{u}$ via $\vec{w} = (x_1, x_2, x_3, x_2^2, x_3)$, since for all $\vec{c} = (e_1, e_2, e_3) \in \mathbb{Z}^3$ we have:

$$\vec{w}(\vec{u}(\vec{c})) = (e_2^2, e_3^3, e_3, e_2^2, e_3) = \vec{u}'(e_1, e_2, e_3, e_2^2, e_3) = \vec{u}'(\vec{w}(\vec{c})).$$

Here, the non-linear part of $\vec{u}$ is mimicked by the variables $y_{x_2}^2$, $y_{x_3}^3$, and $y_{x_3}$.
The linearization of Def. 6.1 also works when applying the update repeatedly.

**Corollary 6.3** (Iterated Update of Linearization). Let \( \vec{u} \in (S[\vec{x}])^d \) and \( \vec{u}' \in (S[\vec{y}])^d \) be its linearization via \( \vec{w} \in (S[\vec{x}])^d \). Then for all \( \vec{c} \in S^d \) and all \( n \in \mathbb{N} \) we have \( \vec{w}(\vec{w}^n(\vec{c})) = (\vec{u}')^n(\vec{w}(\vec{c})) \).

We now define the linearization of a loop to be a linearization of its update where the loop guard is extended to ensure that the fresh variables \( y_{d+1}, \ldots, y_d \) indeed correspond to \( w_{d+1}, \ldots, w_d \).

**Definition 6.4** (Linearization of a Loop). Let \( (\phi, \vec{u}) \) be a loop on \( S^d \) using the variables \( \vec{x} \). A loop \( (\phi', \vec{u}') \) on \( S^d \) using the variables \( \vec{y} \) is a linearization of \( (\phi, \vec{u}) \) via \( \vec{w} \in (S[\vec{x}])^d \) if both
(a) \( \vec{u}' \) is a linearization of \( \vec{u} \) via \( \vec{w} \)
(b) \( \phi' = \phi[x_1/y_1, \ldots, x_d/y_d] \land \land_{i=d+1}^d (y_{w_i} - x_i/y_i, \ldots, x_d/y_d) = 0 \).

**Example 6.5.** Consider the loop \( (\phi, \vec{u}) \) on \( \mathbb{Z}^3 \) where \( \phi = x_2 > x_3 \) and \( \vec{u} = (x_2^2, x_3^2, x_3) \). Then the linearization of \( (\phi, \vec{u}) \) via \( \vec{w} \) is \( (\phi', \vec{u}') \) where \( \vec{u}' \) is as in Ex. 6.2 and \( \phi' \) is
\[
\phi'[x_1/y_1, x_2/y_2, x_3/y_3] \land y_{2x_2} - y_{x_2}^2 = 0 \land y_{x_2}^2 - y_{2x_2}^2 = 0 \land y_{x_2}^3 - y_{2x_2} = 0
= y_{2x_2} > y_{x_2} \land y_{x_2}^2 - y_{2x_2}^2 = 0 \land y_{x_2}^3 - y_{2x_2} = 0 \land y_{x_2}^4 - y_{2x_2} = 0
\]

To illustrate the correspondence between \( (\phi, \vec{u}) \) and \( (\phi', \vec{u}') \), consider the initial value \( \vec{c} = (1, 3, 2) \). Here, the original loop yields the trace \( (\vec{c}, \vec{u}(\vec{c})), (\vec{u}(\vec{c}'), \vec{u}'(\vec{c'})), \ldots) = ((1,3,2), (9,4,2), (16,4,2), \ldots) \). The linearized loop operates over the variables \( (y_1, y_2, x_1, x_2, x_3, y_2^2, y_2^3, y_3^2, y_3^3) \). Thus, the first three variables correspond to \( x_1, x_2, x_3 \) and the latter ones correspond to \( x_2^2, x_3^2, x_3^3 \). So the corresponding initial value is \( \vec{c}' = (1, 3, 2, 9, 4, 16) \) and the resulting trace is \( (\vec{c}', \vec{u}'(\vec{c}')), (\vec{u}'(\vec{c}'), \ldots) = ((1,3,2,9,4,16), (9,4,2,16,4,16), (16,4,2,16,4,16), \ldots) \).

**Lemma 6.6.** Let \( (\phi', \vec{u}') \) on \( S^d \) be a linearization of \( (\phi, \vec{u}) \) on \( S^d \) via \( \vec{w} \).
(a) \( (\phi', \vec{u}') \) terminates on \( \vec{c}' \in S^d \) if there is no \( \vec{c}' \in S^d \) such that \( \vec{c}' = \vec{w}(\vec{c}) \).
(b) The loop \( (\phi, \vec{u}) \) terminates on \( \vec{c}' \in S^d \) iff \( (\phi', \vec{u}') \) terminates on \( \vec{w}(\vec{c}) \).

While Lemma 6.6 proves the soundness of linearization, we now show how to find \( \vec{u}' \) and \( \vec{u} \) automatically, where it suffices to only use monomials (instead of arbitrary polynomials) in \( \vec{w} \). A monomial over \( \vec{x} \) has the form \( x_1^{z_1} \cdot \ldots \cdot x_d^{z_d} \) with \( z_i \in \mathbb{N} \) for all \( 1 \leq i \leq d \). Let \( \vec{z} = (z_1, \ldots, z_d) \) abbreviate \( x_1^{z_1} \cdot \ldots \cdot x_d^{z_d} \).

The original update \( \vec{u} \) consists of polynomials \( u_i \) to update the variable \( x_i \), for all \( 1 \leq i \leq d \). The linearized update \( \vec{u}' \) consists of polynomials \( u_m' \) to update the variables \( y_m \) for all monomials \( m \) in \( \vec{w} \). Here, for any monomial \( m = x_1^{z_1} \cdot \ldots \cdot x_d^{z_d} \), the polynomial \( u_m' \) results from replacing each monomial \( p \) in \( u_1^{z_1} \cdot \ldots \cdot u_d^{z_d} \) (i.e., the monomial \( p \) in each addend \( c \cdot p \) of the polynomial \( u_1^{z_1} \cdot \ldots \cdot u_d^{z_d} \)) by the variable \( y_p \). More precisely, if \( u_1^{z_1} \cdot \ldots \cdot u_d^{z_d} \) has the form
Algorithm 3: Linearizing run-Loops

Input: run-loop \((\varphi, \vec{u})\) using the variables \(\vec{x}\)
Output: linear-update run-loop \((\varphi', \vec{u'})\) and \(\vec{w}\)

such that \((\varphi', \vec{w})\) is a linearization of \((\varphi, \vec{u})\) via \(\vec{w}\)

\[
\begin{align*}
1 & \quad \vec{v} \leftarrow (x_1, \ldots, x_d) \quad \text{and} \quad \vec{w} \leftarrow (x_1, \ldots, x_d) \\
2 & \quad \textbf{while } \vec{v} \neq () \textbf{ do} \\
3 & \quad \quad \text{remove the first monomial } m = x_1^{e_1} \cdots x_d^{e_d} \text{ from } \vec{v} \text{ and insert } m \text{ at the end of } \vec{w} \\
4 & \quad \quad \vec{u}_m' \leftarrow u_1^{e_1} \cdots u_d^{e_d}, \text{ where the monomials } p \text{ in non-constant addends of} \\
5 & \quad \quad \quad \vec{u}_1^{e_1} \cdots \vec{u}_d^{e_d} \text{ are replaced by } y_p \text{ and we insert } p \text{ at the end of } \vec{v} \text{ if } p \text{ is not yet} \\
6 & \quad \quad \quad \text{contained in } \vec{w} \\
7 & \quad \textbf{return } \vec{u} \text{ and } (\varphi[x_1/y_1, \ldots, x_d/y_{ed}] \land \bigwedge_{m = \vec{w} \text{ contains monomial } \vec{v}} (y_m - \prod_{i=1}^{d} y_{e_i}^i = 0), \vec{u'})
\end{align*}

Example 6.7. We apply Alg. 3 to linearize the loop \((\varphi, \vec{u})\) from Ex. 6.5 where \(\varphi\) is \(x_2 > x_3\) and \(\vec{u} = (x_2^2, x_3^2, x_3)\). In the beginning, we have \(\vec{v} = (x_1, x_2, x_3)\).

We start with \(x_1\) and remove it from \(\vec{v}\). In \(u_1 = x_2^2\) we have to replace the monomial \(x_2^2\) by the fresh variable \(y_{x_2}\) when constructing \(u'_{x_1}\). Hence, \(u'_{x_1} = y_{x_2}\) and we obtain \(\vec{v} = (x_2, x_3, x_2^2)\).

Next, we consider \(x_2\), where \(u_2 = x_2^2\). Thus, we obtain \(u'_{x_2} = y_{x_3}\) and \(\vec{v} = (x_2, x_2^2, x_2^3)\). Then we take \(x_3\), where \(u_3 = x_3\). Hence, \(u'_{x_3} = y_{x_3}\), but we do not insert \(x_3\) into \(\vec{v}\) again, since we just computed \(u'_{x_3}\). So we have \(\vec{v} = (x_2, x_3, x_2^3)\).

We now handle \(x_2^2\). For the linearized update, we take \(u_2' = x_2^2\) but replace the monomial \(x_2^2\) by a fresh variable \(y_{x_2}\). Hence, \(u'_{x_2} = y_{x_3}\) and \(\vec{v} = (x_2^2, x_3)\).

Next, we take the monomial \(x_2^3\) and in \(u_3' = x_3\) we have to replace the monomial by \(y_{x_3}\). This leads to the linearized update \(u'_{x_3} = y_{x_3}\), but we do not insert \(x_3\) into \(\vec{v}\) again, since we just computed \(u'_{x_3}\). Hence, we have \(\vec{v} = (x_2^4)\).

Finally, we treat \(x_3\) and in \(u_3' = x_3\) we replace the monomial by \(y_{x_3}\), i.e., \(u'_{x_3} = y_{x_3}\). Now \(\vec{v} = ()\). Hence, Alg. 3 terminates and returns the loop with the guard \(\varphi'\) from Ex. 6.5 and the (linear) update \((y_{x_2}, y_{x_3}, y_{x_3}, y_{x_3}, y_{x_3}, y_{x_3})\) over the variables \((y_{x_1}, y_{x_2}, y_{x_2}, y_{x_2}, y_{x_3}, y_{x_3})\), as in Ex. 6.2.

Now we infer an upper bound on Alg. 3’s complexity. To this end, we will show that the degrees of the monomials in \(\vec{v}\) which are used for the linearization can be bounded by the maximal dependency degree of the loop’s update \(\vec{u}\). For \(1 \leq i \leq d\), the dependency degree \(\text{depdeg}_\vec{u}(x_i)\) is the degree of
states three easy observations on \( \text{depdeg} \). Here, let \( \mathbf{u} \) be a \( \mathbf{twn} \)-loop with \( u \), but this degree is expressed in terms of those variables that are minimal w.r.t. \( \succ_\mathbf{u} \). Recall that \( u \) has the form \( c_i \cdot x_i + p_i \) where \( p_i \) is a polynomial which only contains variables that are smaller than \( x_i \) w.r.t. \( \succ_\mathbf{u} \). W.l.o.g. we may assume that \( x_i \succ_\mathbf{u} x_j \) implies \( i > j \) for all \( 1 \leq i, j \leq d \). Then for every monomial \( x_i^{z_i+1} \cdots x_d^{z_d} \) in \( p_i \), the corresponding dependency degree is \( z_i+1 \cdot \text{depdeg}_\mathbf{u}(x_{i+1}) + \cdots + z_d \cdot \text{depdeg}_\mathbf{u}(x_d) \). The dependency degree of \( p_i \) is the maximal dependency degree of its monomials.

**Definition 6.8** (Dependency Degree). Let \( (\varphi, \bar{u}) \) be a \( \mathbf{twn} \)-loop with \( u = c_i \cdot x_i + p_i \) for all \( 1 \leq i \leq d \), where \( p_i \) only contains variables that are smaller than \( x_i \) w.r.t. \( \succ_\mathbf{u} \). We define the dependency degree w.r.t. \( \bar{u} \) as follows:

- \( \text{depdeg}_\mathbf{u}(x_i) = \max\{1, \text{depdeg}_\mathbf{u}(p_i)\} \) for all \( 1 \leq i \leq d \).
- \( \text{depdeg}_\mathbf{u}(p) = \max\{\text{depdeg}_\mathbf{u}(m) \mid m \text{ is a monomial in } p\} \) for every non-zero \( p \in S[\bar{x}] \) and \( \text{depdeg}_\mathbf{u}(0) = -\infty \).
- \( \text{depdeg}_\mathbf{u}(x_1^{z_1} \cdots x_d^{z_d}) = \sum_{i=1}^d z_i \cdot \text{depdeg}_\mathbf{u}(x_i) \) for all \( z_1, \ldots, z_d \in \mathbb{N} \).

Since \( \succ_\mathbf{u} \) is well founded by the triangularity of the loop, \( \text{depdeg}_\mathbf{u} \) is well defined: for the variables \( x_1 \) which are minimal w.r.t. \( \succ_\mathbf{u} \), \( p_i \) is a constant and thus, \( \text{depdeg}_\mathbf{u}(x_1) = 1 \). For other variables \( x_i \) with \( p_i \neq 0 \), we can compute \( \text{depdeg}_\mathbf{u}(p_i) \) because \( \text{depdeg}_\mathbf{u}(x_j) \) is already known for all variables \( x_j \) occurring in \( p_i \). Lemma 6.9 states three easy observations on \( \text{depdeg} \). Here, \( \text{deg} \) denotes the degree of monomials or polynomials, i.e., \( \text{deg}(x_1^{z_1} \cdots x_d^{z_d}) = z_1 + \cdots + z_d \).

**Lemma 6.9.** Let \( (\varphi, \bar{u}) \) be a \( \mathbf{twn} \)-loop.

(a) For every monomial \( m \) over \( \bar{x} \), we have \( \text{deg}(m) \leq \text{depdeg}_\mathbf{u}(m) \).

(b) If \( \text{deg} \) is the maximum of 1 and the highest degree of any polynomial in \( \bar{u} \), then for any \( 1 \leq i \leq d \) we have \( \text{depdeg}_\mathbf{u}(x_i) \leq \text{deg}^{d-i} \).

(c) For \( \text{mdepdeg} = \max\{\text{depdeg}_\mathbf{u}(x_i) \mid 1 \leq i \leq d\} \), we have \( \text{mdepdeg} \leq \text{deg}^{d-1} \).

**Example 6.10.** Again, we consider a loop as in Ex. 6.7 with update \( \bar{u} = (x_2^2, x_3^2, x_2 x_3) \). Then, \( \text{depdeg}_\mathbf{u}(x_3) = \max\{1, \text{depdeg}_\mathbf{u}(0)\} = \max\{1, -\infty\} = 1 \), \( \text{depdeg}_\mathbf{u}(x_2) = \text{depdeg}_\mathbf{u}(x_2^2) = 2 \cdot \text{depdeg}_\mathbf{u}(x_3) = 2 \cdot 1 = 2 \), and \( \text{depdeg}_\mathbf{u}(x_1) = \text{depdeg}_\mathbf{u}(x_2^2) = 2 \cdot \text{depdeg}_\mathbf{u}(x_2) = 2 \cdot 2 = 4 \). So intuitively, the update of \( x_1 \) has degree 4 in terms of the \( \succ_{\bar{u}} \)-minimal variable \( x_3 \) since the update of \( x_2 \) is quadratic in \( x_3 \) and \( x_2^2 \) then has degree 4 w.r.t. \( x_3 \). Here, \( \text{mdepdeg} = 4 \) and for the maximal degree \( \text{deg} = 2 \) occurring in the update, we indeed have \( \text{mdepdeg} = \text{deg}^{d-1} = 2^{d-1} \). So the bound on \( \text{mdepdeg} \) in Lemma 6.9 (c) is tight.

As another example, consider the update \( \bar{u} = (3 \cdot x_1 + 5 \cdot x_2^4 \cdot x_3^6 + 7 \cdot x_3^8, x_2^2, 9) \). Now we have \( \text{depdeg}_\mathbf{u}(x_3) = \max\{1, \text{depdeg}_\mathbf{u}(9)\} = \max\{1, 0\} = 1 \), \( \text{depdeg}_\mathbf{u}(x_2) = \text{depdeg}_\mathbf{u}(x_2^2) = 2 \cdot \text{depdeg}_\mathbf{u}(x_3) = 2 \cdot 1 = 2 \), and \( \text{depdeg}_\mathbf{u}(x_1) = \text{depdeg}_\mathbf{u}(5 \cdot x_2^4 \cdot x_3^6 + 7 \cdot x_3^8) = \max\{\text{depdeg}_\mathbf{u}(x_2^4 \cdot x_3^6), \text{depdeg}_\mathbf{u}(x_3^8)\} = \max\{4 \cdot \text{depdeg}_\mathbf{u}(x_2), 6 \cdot \text{depdeg}_\mathbf{u}(x_3), 8 \cdot \text{depdeg}_\mathbf{u}(x_3)\} = \max\{4 \cdot 2 + 6 \cdot 1, 8 \cdot 1\} = 14 \).

Now we prove that Alg. 3 only constructs updates \( u_i' \) for monomials \( m \) with \( \text{depdeg}(m) \leq \text{mdepdeg} \). Hence, this also proves termination of the algorithm since there are only finitely many such monomials, and it allows us to give a bound on the number of iterations of the algorithm’s \( \text{while} \)-loop.
Theorem 6.11 (Dependency Degree Suffices for Linearization).
(a) Alg. 3 only computes \( u'_m \) for monomials \( m \) with \( \text{depdeg}(m) \leq \text{mdepdeg} \).
(b) The while-loop of Alg. 3 is executed at most \( \binom{d + \text{mdepdeg}}{\text{mdepdeg}} - 1 \) times.
(c) Alg. 3 terminates.

The following theorem summarizes the main properties of Alg. 3.

Theorem 6.12 (Soundness of Alg. 3).
(a) For any tnn-loop \((\varphi, \vec{u})\), Alg. 3 computes a linearization \((\varphi', \vec{u}')\) via \( \vec{w} = (n_1, \ldots, n_d) \), where \( \deg(n_i) \leq \text{depdeg}(m_i) \leq \text{mdepdeg} \) for all \( 1 \leq i \leq d' \).
(b) The loop \((\varphi, \vec{u})\) terminates iff \((\varphi', \vec{u}')\) does.
(c) The loop \((\varphi', \vec{u}')\) is a linear-update tnn-loop.

As mentioned, the technique in this section is based on the linearization method of [47], where instead of tnn-loops as in Thm. 6.12, [47] works in the setting of solvable loops (Def. 2.3). But [47] has no notion like the dependency degree of Def. 6.8. Instead they only consider the degree of the polynomials in the update \( \vec{u} \). However, Ex. 6.7 shows that the polynomials in \( \vec{w} \) that are used for the linearization may have a higher degree than the ones in \( \vec{u} \). Here, the polynomials in \( \vec{u} = (x_2^3, x_3^2, x_3) \) only have degree 2. However, \( x_1 \) is (eventually) updated to \( x_3^3 \). Thus, to linearize this loop, polynomials up to degree 2 do not suffice, but \( \vec{w} \) must contain a polynomial of degree 4 like \( x_3^4 \).

As we showed in Thm. 6.11 (a), the dependency degree (and hence, also the degree) of the polynomials in \( \vec{u} \) is bounded by \( \text{mdepdeg} = \max\{\text{depdeg}_\vec{u}(x_i) | 1 \leq i \leq d\} \). Indeed, in Ex. 6.7 we have \( \text{depdeg}_\vec{u}(x_1) = 4 \). Hence, our new concept of the dependency degree was needed for the upper bound on the number of iterations of the linearization algorithm in Thm. 6.11 (b). Based on this, we can now infer the asymptotic complexity of Alg. 3. As mentioned, we will need this in Sect. 7.2 to analyze the complexity of deciding termination of tnn-loops.

By Thm. 6.11 (b), the while-loop of Alg. 3 is executed at most \( \binom{d + \text{mdepdeg}}{\text{mdepdeg}} - 1 \) times. Since \( \binom{a}{b} \in \mathcal{O}(n^b) \) for any natural numbers \( n \geq k \), we have

\[
\binom{d + \text{mdepdeg}}{\text{mdepdeg}} - 1 = \binom{d + \text{mdepdeg}}{d} \in \mathcal{O}((d + \text{mdepdeg})^{\text{mdepdeg}}) \cap \mathcal{O}((d + \text{mdepdeg})^d).
\]

By Lemma 6.9 (c), we have \( \text{mdepdeg} \leq \text{deg}^{d-1} \) where \( \text{deg} \) is the maximum of 1 and the highest degree of any polynomial in the update \( \vec{u} \). Hence,

\[
\binom{d + \text{mdepdeg}}{d} \in \mathcal{O}((d + \text{deg}^{d-1})^d) \cap \mathcal{O}((d + \text{deg}^{d-1})^\text{deg}^{d-1}) \subseteq \mathcal{O}((d + \text{deg}^{d-1})^d).
\]

For the expression \( (d + \text{deg}^{d-1})^d \) we have (see also App. A.23):

\[
(d + \text{deg}^{d-1})^d \leq 2 \cdot 2^{d+1d\text{ld}(\text{deg})\cdot(d-1)\cdot d}
\]

(15)

Here, as usual, \( \text{ld} \) denotes the logarithm to the base 2.

Thus, \( (d + \text{deg}^{d-1})^d \) is at most exponential in \( d \), i.e., the number of iterations of Alg. 3 is at most exponential in \( d \). In each such iteration, one has to compute a new polynomial \( u_1^a \cdot \ldots \cdot u_d^a \). By Thm. 6.11 (a), this polynomial only
contains monomials \( m \) with \( \text{depdeg}(m) \leq \text{mdpdeg} \) and there are \((d + \text{mdpdeg}) \in \mathcal{O}(\text{deg}^{d-1}) \) many such monomials (see Thm. 6.11(b)). To compute their coefficients, one has to multiply up to \( z_1 + \ldots + z_d \) factors, where \( z_1 + \ldots + z_d \leq \text{mdpdeg} \leq \text{deg}^{d-1} \). This corresponds to a nested multiplication of two factors, where the result of one multiplication step is the input to the next multiplication, and the depth of the nesting is exponential in \( d \). So the results and the factors of the multiplications grow at most doubly exponential in \( d \). Therefore, this proves Lemma 6.13(a), i.e., the runtime of Alg. 3 is at most double exponential.

However, if the number of variables \( d \) is bounded by a constant \( D \), then the number of iterations of Alg. 3 and the number of monomials in the linearized updates is bounded by \((d + \text{mdpdeg}) \in \mathcal{O}(\text{deg}^D) \), which is polynomial in \( \text{deg} \). For their coefficients, one has to multiply up to \( \text{mdpdeg} \) factors, i.e., this corresponds to a nested multiplication where the depth of the nesting is polynomial in \( \text{deg} \). So the results and the factors of the multiplications grow at most exponentially in \( \text{deg} \). Therefore, then linearization can be computed in exponential time. This proves Lemma 6.13(b).

**Lemma 6.13.** Let \( D \in \mathbb{N} \) be fixed. The linearization of a tnn-loop

(a) can be computed in double exponential time.

(b) can be computed in exponential time if the number of variables \( d \) is at most \( D \).

**Example 6.14.** While Lemma 6.13 only gives upper bounds on the complexity of linearization, the loop \( L_{\text{non-pspace}} \) from (10) can be used to infer lower bounds. Here, the linearized loop operates on the variables

\[
y_{x_1}, \ldots, y_{x_d}, \ y_{x_2}, \ldots, y_{x_d}, \ y_{x_3}, \ldots, y_{x_d}, \ldots, y_{x_d}, y_{x_d}, y_{x_d}
\]

and the corresponding linearized update \( \vec{u'} \) instantiates

- \( y_{x_1} \) by \( y_{x_1} \),
- \( y_{x_2} \) with \( y_{x_3} \left( \text{deg} + 1 \right) \) for all \( 0 \leq i \leq 1 \),
- \( \ldots \)
- \( y_{x_d} \) with \( y_{x_{d-1}} \left( \text{deg} - 2 \right) \) for all \( 0 \leq i \leq d - 2 \), and
- \( y_{x_d} \) with \( \text{deg} \cdot y_{x_{d-1}} \) for all \( 0 \leq i \leq d - 1 \).

So in particular, the update contains the constant \( d^{d-1} \) which shows that this linearization requires exponential space if \( d \) is not bounded.

### 7 Complexity of Deciding Termination

In this section, we study the complexity of deciding termination for different classes of loops by using our results from Sect. 4 to 6. We first regard linear-update loops in Sect. 7.1, where the update is of the form \( \vec{x} \leftarrow A \cdot \vec{x} + \vec{b} \) with \( A \in \mathbb{R}^{n \times n} \)....
while ($\varphi$) do $\vec{x} \leftarrow A \cdot \vec{x} + \vec{b}$

while ($\varphi \land x = 1$) do $\begin{bmatrix} \vec{x} \\ x \end{bmatrix} \leftarrow \begin{bmatrix} A & \vec{b} \\ \vec{b}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} \vec{x} \\ x \end{bmatrix}$

Fig. 4: Linear-Update Loop

Fig. 5: Homogeneous Linear-Update Loop

$S^{d \times d}$ and $\vec{b} \in S^d$. The reason for this restriction is that such loops can always be transformed into $twn$-form by our transformation $Tr$ from Sect. 5. More precisely, we show that termination of linear loops with rational spectrum is $\text{Co-NP}$-complete if $S \in \{ \mathbb{Z}, \mathbb{Q}, \mathbb{R}_A \}$ and that termination of linear-update loops with real spectrum is $\forall \mathbb{R}$-complete if $S = \mathbb{R}_A$. Since our proof is based on a reduction to $\text{Th}_3(S, \mathbb{R}_A)$, and $\mathbb{R}_A$ and $\mathbb{R}$ are elementary equivalent, our results also hold if the program variables range over $\mathbb{R}$. By combining these results with our observations on the complexity of linearization from Sect. 6, we then analyze the complexity of deciding termination for arbitrary $twn$-loops in Sect. 7.2. In Sect. 7.3, we show that there is an important subclass of linear loops where our decision procedure for termination works efficiently, i.e., when the number of eigenvalues of the update matrix is bounded, then termination can be decided in polynomial time. Here, we again use our transformation $Tr$ from Sect. 5.

For our complexity results, we assume the usual dense encoding of univariate polynomials, i.e., a polynomial of degree $k$ is represented as a list of $k + 1$ coefficients. As discussed in [54], many problems which are considered to be efficiently solvable become intractable if polynomials are encoded sparsely (e.g., as lists of monomials where each monomial is a pair of its non-zero coefficient and its degree). With densely encoded polynomials, all common representations of algebraic numbers can be converted into each other in polynomial time [3].

7.1 Complexity of Deciding Termination for Linear-Update Loops

When analyzing linear-update loops, w.l.o.g. we can assume $\vec{b} = \vec{0}$ since a loop of the form in Fig. 4 terminates iff the loop in Fig. 5 terminates, where $x$ is a fresh variable (see [30, 48]). Moreover, $\vec{e}$ witnesses (eventual) non-termination for the loop in Fig. 4 iff $\begin{bmatrix} \vec{e} \\ 1 \end{bmatrix}$ witnesses (eventual) non-termination for the loop in Fig. 5. Note that the only eigenvalue of $\begin{bmatrix} A & \vec{b} \\ \vec{b}^T & 1 \end{bmatrix}$ whose multiplicity increases in comparison to $A$ is 1. Thus, to decide termination of linear-update loops with rational or real spectrum, respectively, it suffices to decide termination of homogeneous loops of the form $(\varphi, A \cdot \vec{x})$ where $A$ has only rational or real eigenvalues.

Such loops can always be transformed into $twn$-form using our transformation $Tr$ from Sect. 5. To compute the required automorphism $\eta$, we compute the Jordan normal form $Q$ of $A$ together with the corresponding transformation matrix $T$, i.e., $T$ is an invertible real matrix such that $A = T^{-1} \cdot Q \cdot T$. Then $Q$ is a triangular real matrix whose diagonal consists of the eigenvalues $\lambda \in \mathbb{R}_A$ of $A$. We define $\eta \in \text{End}_{\mathbb{R}_A}(\mathbb{R}_A[\vec{x}])$ by $\eta(\vec{x}) = T \cdot \vec{x}$. Then $\eta \in \text{Aut}_{\mathbb{R}_A}(\mathbb{R}_A[\vec{x}])$.
has the inverse $\eta^{-1}(\vec{x}) = T^{-1} \cdot \vec{x}$. Thus, $Tr_{\eta}(\varphi, A \cdot \vec{x})$ is a \textit{turn}-loop with the update

$$(\eta(\vec{x})) (A \cdot \vec{x}) (\eta^{-1}(\vec{x})) = T \cdot A \cdot T^{-1} \cdot \vec{x} = Q \cdot \vec{x}.$$ 

To analyze termination of the loop on $S^d$, we have to consider termination of the transformed loop on $F = \tilde{\eta}(S^d) = T \cdot S^d$ (see Cor. 5.16).

The Jordan normal form $Q$ as well as the matrices $T$ and $T^{-1}$ can be computed in polynomial time \cite{23, 53}. Hence, we can decide whether all eigenvalues are rational or real numbers in polynomial time by checking the diagonal entries of $Q$. Thus, we obtain the following lemma.

**Lemma 7.1.** Let $(\varphi, A \cdot \vec{x})$ be a linear-update loop.

(a) It is decidable in polynomial time whether $A$ has only rational or real eigenvalues.

(b) If $A$ has only real eigenvalues, we can compute a linear $\eta \in Aut_{\mathbb{R}}(\mathbb{R}[\vec{x}])$ such that $Tr_{\eta}(\varphi, A \cdot \vec{x})$ is a linear-update \textit{turn}-loop in polynomial time.

(c) If $(\varphi, A \cdot \vec{x})$ is a linear loop, then so is $Tr_{\eta}(\varphi, A \cdot \vec{x})$.

So every linear-update loop with real spectrum can be transformed into a linear-update \textit{turn}-loop, i.e., the transformation $Tr$ from Sect. 5 is \textit{complete} for such linear-update loops. Note that Lemma 7.1 (a) yields an efficient check whether a given linear-update loop has rational or real spectrum.

As chaining (Def. 3.1) can clearly be done in polynomial time, w.l.o.g. we may assume that $Tr_{\eta}(\varphi, A \cdot \vec{x}) = (\varphi', Q \cdot \vec{x})$ is $\textit{turn}$. Next, to analyze termination of a $\textit{turn}$-loop, our technique of Sect. 4 (resp. Sect. 5.3) uses a closed form for the update. For $\textit{turn}$-loops $(\varphi', Q \cdot \vec{x})$ where $Q$ is a triangular matrix with non-negative diagonal entries, a suitable (i.e., poly-exponential) closed form can be computed in polynomial time analogously to \cite[Prop. 5.2]{34}. This closed form is linear in $\vec{x}$ (we will discuss this closed form in Sect. 7.3, see Eq. (22)).

According to our approach in Sect. 5.3, we now proceed as in Alg. 2 and construct the formula $\exists \vec{x} \in \mathbb{R}^d_+, \psi_F \land \text{red}(\psi(\vec{q}_\text{norm})) \in Th_3(S, \mathbb{R}_A)$ in polynomial time due to Cor. 5.24. Hence, we get the following lemma.

**Lemma 7.2.** Let $(\varphi, A \cdot \vec{x})$ be a linear-update loop with real spectrum. Then we can compute a closed formula $\psi \in Th_3(S, \mathbb{R}_A)$ in polynomial time such that $\psi$ is valid iff the loop is non-terminating. If $\varphi$ is linear, then so is $\psi$.

Note that $\psi$ is existentially quantified. Hence, if the loop has rational spectrum and coefficients, $\varphi$ (and thus also $\psi$) is linear, and $S \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}_A, \mathbb{R}\}$, then invalidity of $\psi$ is in $\text{Co-NP}$ as validity of such formulas is in $\text{NP}$, see \cite[Thm. 42]{49}. Thus, we get the first main result of this section. Here, we fix an inaccuracy in \cite[Thm. 42]{20}, where we also allowed irrational eigenvalues and coefficients, and thus $\psi$ may contain irrational coefficients. However, to the best of our knowledge, it is not known whether validity of linear formulas from $Th_3(S, \mathbb{R}_A)$ with irrational algebraic coefficients is in $\text{NP}$.

**Theorem 7.3** (Co-NP-Completeness). For linear loops $(\varphi, A \cdot \vec{x} + \vec{b})$ with rational spectrum where $\varphi \in Th_{df}(\mathbb{Q}), A \in \mathbb{Q}^{d \times d}$, and $\vec{b} \in \mathbb{Q}^d$, termination over $\mathbb{Z}, \mathbb{Q}, \mathbb{R}_A$, and $\mathbb{R}$ is $\text{Co-NP}$-complete.
Co-NP-hardness follows from Co-NP-hardness of unsatisfiability of propositional formulas: let $\xi$ be a propositional formula over the variables $\vec{x}$. Then the loop $(\xi[x_i / (x_i > 0)] \mid 1 \leq i \leq d, \vec{x})$ terminates iff $\xi$ is unsatisfiable.

We now consider linear-update loops with real spectrum (and possibly non-linear conditions) on $\mathbb{R}_A^d$ and $\mathbb{R}^d$. Here, non-termination is $\exists \mathbb{R}$-complete.

**Definition 7.4 ($\exists \mathbb{R}$ [56, 57]).** Let

$$Th_3(\mathbb{R})^\top = \{ \psi \in Th_3(\mathbb{R}) \mid \psi \text{ closed and valid} \}.$$ 

The complexity class $\exists \mathbb{R}$ is the closure of $Th_3(\mathbb{R})^\top$ under poly-time-reductions.

We have $NP \subseteq \exists \mathbb{R} \subseteq PSPACE$ (see [12]). By Lemma 7.2, non-termination of linear-update loops over $\mathbb{R}_A$ with real spectrum is in $\exists \mathbb{R}$. It is also $\exists \mathbb{R}$-hard since $(\phi, \vec{x})$ is non-terminating iff $\exists \vec{x} \in \mathbb{R}_A^d, \phi$ is valid. So non-termination is $\exists \mathbb{R}$-complete, i.e., termination is Co-$\exists \mathbb{R}$-complete (where Co-$\exists \mathbb{R} = \forall \mathbb{R}$ [57]).

**Theorem 7.5 ($\forall \mathbb{R}$-Completeness).** Termination of linear-update loops with real spectrum over $\mathbb{R}_A$ and $\mathbb{R}$ is $\forall \mathbb{R}$-complete.

### 7.2 Complexity of Deciding Termination for $twn$-Loops over $S \in \{ \mathbb{R}_A, \mathbb{R} \}$

By Cor. 4.11 (b) termination is decidable for arbitrary $twn$-loops over $\mathbb{R}_A$ and $\mathbb{R}$. So in this section, we discuss the complexity of this decision problem. First of all, deciding termination of arbitrary $twn$-loops over $\mathbb{R}_A$ and $\mathbb{R}$ is $\forall \mathbb{R}$-hard since termination of linear-update loops with real spectrum over $\mathbb{R}_A$ and $\mathbb{R}$ is $\forall \mathbb{R}$-hard by Thm. 7.5 and any such linear-update loop can be transformed into a $twn$-loop in polynomial time by Lemma 7.1 (b). Thus, we have a lower bound on the complexity of deciding termination for arbitrary $twn$-loops.

To find an upper asymptotic bound for deciding termination of $twn$-loops, first note that we can restrict ourselves to $tnn$-loops again, as any $twn$-loop can be transformed into a $tnn$-loop by chaining (Def. 3.1) in polynomial time.

The next step of our decision procedure in Alg. 1 is to compute closed forms for the update. However, the loop $L_{non-pspace}$ in (10) showed that in general, the computation of closed forms cannot be done in polynomial space. On the other hand, as mentioned in Sect. 7.1, for linear-update loops, closed forms can be computed in polynomial time. To benefit from this upper bound, we therefore do not proceed directly as in Alg. 1, but instead we first linearize the $tnn$-loop. While linearization cannot be computed in polynomial space either (see Ex. 6.14), in Sect. 6 we formalized and analyzed the complexity of the linearization technique from [47]. Given a $tnn$-loop $(\phi, \vec{u})$ we can compute a linear-update $tnn$-loop $(\phi', \vec{u}')$ such that $(\phi, \vec{u})$ terminates if and only if $(\phi', \vec{u}')$ does (Thm. 6.12). Clearly, linear-update $tnn$-loops over $\mathbb{R}_A$ and $\mathbb{R}$ always have real spectrum as the eigenvalues of a triangular matrix are its diagonal entries.

So by using this linearization, we can give an upper complexity bound for deciding termination of arbitrary $twn$-loops.

As linearization can be computed in double exponential time and thus, also in double exponential space, and termination of linear-update loops is in $\forall \mathbb{R} \subseteq$
PSPACE ⊆ EXPTIME by Lemma 7.2 (where the size of the linear-update loop may be at most doubly exponential), we obtain that deciding termination of $\text{tv}-$loops is in $3\text{-EXPTIME}$ (i.e., it is between $\forall \mathbb{R}$ and $3\text{-EXPTIME}$). Moreover, if the number of variables is bounded, checking validity of a formula in $\text{Th}_3(\mathbb{R}_A)$ is in $\mathbb{P}$ (see [2]). In this case, combining linearization which can be computed in exponential time and thus, also in exponential space when $d$ is bounded (Lemma 6.13 (b)), and deciding termination of linear-update loops which is polynomial in this case (where the size of the linear-update loop may be at most exponential), we obtain Thm. 7.6 (b).

**Theorem 7.6** (Membership in $3\text{-EXPTIME}$). Let $D \in \mathbb{N}$ be fixed. Termination of $\text{tv}$-loops over $\mathbb{R}_A$ and $\mathbb{R}$

(a) is in $3\text{-EXPTIME}$.

(b) is in $\text{EXPTIME}$ if the number of variables $d$ is at most $D$.

### 7.3 Complexity of Deciding Termination for Uniform Loops

In Sect. 7.1, we showed that termination of linear loops with rational spectrum is Co-NP-complete. For proving Co-NP-hardness, we used the trivial update $\vec{x} \leftarrow \vec{x}$ induced by the identity matrix. Therefore, the question arises whether imposing suitable restrictions to the update matrix (which exclude the identity matrix) leads to a “more efficient” decision procedure for termination (assuming $\mathbb{P} \neq \mathbb{NP}$). We now analyze a special case of linear loops (so-called uniform loops) and show that for these loops deciding termination is polynomial, if one fixes the number of eigenvalues of the update matrix.

In Sect. 7.3.1, we introduce uniform loops and parameterized decision problems, and state the main result of Sect. 7.3 (Thm. 7.10). To prove it, we show that for uniform loops, instantiating the variables in the loop guard by $\vec{q}_{\text{norm}}$ (as required by our decision procedure from Sect. 4) results in formulas of a special structure (so-called interval conditions, see Sect. 7.3.2 and 7.3.3). Validity of these formulas can be checked in polynomial time (Sect. 7.3.4) which proves Thm. 7.10 for uniform loops over $\mathbb{Q}$, $\mathbb{R}_A$, and $\mathbb{R}$. In Sect. 7.3.5 we show that our result holds for uniform loops over $\mathbb{Z}$ as well.

#### 7.3.1 Uniform Loops and the Parameterized Complexity Class XP

**Definition 7.7** (Uniform Loop). A linear loop $(\varphi, A \cdot \vec{x})$ over $S \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}_A\}$ is uniform if each eigenvalue $\lambda$ of $A$ is a non-negative number from $S$ whose eigenspace w.r.t. $A$ is one-dimensional, i.e., $\lambda$ has geometric multiplicity 1.

The latter property is equivalent to requiring that there is exactly one Jordan block for each eigenvalue in $A$’s Jordan normal form. To grasp uniform loops intuitively, consider triangular linear loops with updates $x_i \leftarrow \lambda \cdot x_i + p_i$ for all $1 \leq i \leq d$, where the factor $\lambda \geq 0$ is the same for all $i$. These loops are uniform iff the relation $\succ_{\vec{a}}$ is total (or equivalently, iff the variables can be ordered such that the super-diagonal of $A$ does not contain zeros).
while \((x_1 \geq -x_3)\) do
\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} \leftarrow \begin{bmatrix} x_1 + x_2 \\ x_2 + x_3 \\ x_3 - 1 \end{bmatrix}
\]

while \((x_1 \geq -x_3 \land x_4 = 1)\) do
\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix} \leftarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
\]

Fig. 6: Uniform Loop [7] via Polynomials

Fig. 7: Uniform Loop [7] via Matrix

Lemma 7.8. A triangular linear loop \((\varphi, A \cdot \vec{x})\) where all diagonal entries are identical and non-negative is uniform iff \(\succ \vec{u}\) is a total ordering.

Thus, loops like the leading example from [7] in Fig. 6 which is equivalent to Fig. 7 are uniform. In contrast, a loop is not uniform if each \(x_i\) is updated to \(\lambda x_i + c_i\) for constants \(c_i \in S\). The reason is that the \(x_i\) do not occur in each other’s updates. Hence, we have \(x_i \not\succ x_j\) and \(x_j \not\succ x_i\) for all \(1 \leq i, j \leq d\).

So in particular, a uniform loop cannot have more than one update of the form \(x_i \leftarrow x_i\). However, the loop condition can still be an arbitrary Boolean formula over linear inequations. Thus, our complexity result is quite surprising since it shows that for this class of loops, termination is easier to decide than satisfiability of the condition (e.g., unsatisfiability of linear formulas over \(\mathbb{R}_a\) is Co-NP-complete). Intuitively, the reason is that our class prohibits multiple updates like \(x_i \leftarrow x_i\) where variables “stabilize” and where termination is essentially equivalent to unsatisfiability of the condition.

To give an intuition how hard the restriction to uniform loops is, we analyzed the TPDB [62] used at the Termination and Complexity Competition [25]. In the category for “Termination of Integer Transition Systems (ITSs)“ we identified 467 polynomial loops with non-constant guard (i.e., termination is not trivial) and 290 (62 \%) of them are uniform loops over \(\mathbb{Z}\). Similarly, in the category for “Complexity Analysis of ITSs“ we found 1,258 such polynomial loops and 452 (36 \%) are uniform. In fact, in practice one is often interested in termination of triangular loops where after chaining, all variables belong to the eigenvalues 0 or 1. The reason is that termination is usually easy to show if there are eigenvalues greater than 1, because they lead to exponential growth. Thus, if the loop terminates, termination is usually reached after few steps (e.g., consider a loop \((x \leq c, 2 \cdot x)\) for any constant \(c\)). Hence, the number of eigenvalues \(k\) is usually smaller than the number \(d\) of program variables. In this section we show that the complexity for deciding termination of uniform loops is exponential in \(k\) but not in \(d\). More precisely, termination of uniform loops is in the parameterized complexity class \(XP\), where the parameter is the number \(k\) of eigenvalues.

Definition 7.9 (Parameterized Decision Problem, \(XP\) [16]). A parameterized decision problem is a language \(L \subseteq \Sigma^* \times \mathbb{N}\), where \(\Sigma\) is a finite alphabet. The second component (from \(\mathbb{N}\)) is called the parameter of the problem.

A parameterized problem \(L\) is slicewise polynomial if the time needed for deciding the question “\((x, k) \in L?\)” is in \(O(|x|^{f(k)})\) where \(f\) is a computable function depending only on \(k\). The corresponding complexity class is called \(XP\).

In the remainder of this section, we prove that for any fixed \(k \in \mathbb{N}\), termination of uniform loops with \(k\) eigenvalues is decidable in polynomial time.
Here, for any \( \alpha \in \mathbb{Q} \), \( \deg_{x_i}(\alpha) \) is the highest power of \( x_i \) occurring in a monomial of \( \alpha \), i.e., it is the degree of \( \alpha \) when interpreting all variables besides \( x_i \) as constants.

### 7.3.2 Hierarchical Expressions and Partitions

We now elaborate on the closed forms arising from uniform loops. To this end, we fix a uniform loop \((\varphi, A \cdot \vec{x})\) where \( A \) is an \( n \times n \) matrix with \( \lambda_1, \ldots, \lambda_k \) being its eigenvalues and \( \beta \) being the dimension of \( A \)'s Jordan blocks. Let \( Q = \text{diag}(Q_{\lambda_1}, \ldots, Q_{\lambda_k}) \) be the transformation matrix defined by \( \eta(\vec{x}) = Q \cdot \vec{x} \) and \( \varphi \in \mathbb{R}_k \), where \( \eta(\vec{x}) = (\varphi, Q \cdot \vec{x}) \) as in Sect. 4.

Instead of termination of the original loop on \( S^n \), we now have to prove termination of the transformed loop on \( \eta(S^n) = T \cdot S^n \). For \( S \in \{Q, R_k\} \) and \( k \) being the multiplicity of \( \eta(\vec{x}) \) over \( S \). In contrast, if \( S = \mathbb{Z} \), then even if the eigenvalues of \( A \) are integers, the transformation matrix \( T \) or its inverse may contain non-integer rational numbers. Thus, we first regard uniform loops over \( S \in \{Q, R_k\} \) and discuss the case \( S = \mathbb{Z} \) in Sect. 7.3.5.

By uniformity, \( Q = \text{diag}(Q_{\lambda_1}, \ldots, Q_{\lambda_k}) \) has \( k \) Jordan blocks \( Q_{\lambda_1}, \ldots, Q_{\lambda_k} \) where \( Q_{\lambda_i} \) is as in Fig. 8. For each eigenvalue \( \lambda_i \), let \( \nu(\lambda_i) \) be the dimension of \( Q_{\lambda_i} \). Since each eigenvalue has geometric multiplicity \( 1 \), \( \nu(\lambda_i) \) is the algebraic multiplicity of \( \lambda_i \) and \( \nu(\lambda) \) is the root of the characteristic polynomial of \( A \). For \( \nu = \nu(\lambda) \), Fig. 9 shows the form of \( Q_{\lambda} \) as in, e.g., [34, 48], where \( \binom{n}{s} = 0 \) if \( n < s \). This directly yields a closed form \( \tilde{q} \) for the \( n \)-fold application of the update \( Q \cdot \vec{x} \) of the transformed loop. Since our approach from Sect. 4 works by analyzing eventual non-termination, we are only interested in validity of formulas for large enough \( n \). Thus, we may assume that \( n \) is larger than the algebraic multiplicities \( \nu(\lambda) \) of all eigenvalues \( \lambda \in \text{spec}(A) \). Then one obtains a resulting normalized closed form \( \tilde{q}_{\text{norm}} \) which consists of normalized polynomial expressions of a special form, so-called hierarchical expressions. Here, for any \( \alpha \in \mathbb{Q}[\vec{x}], \deg_{x_i}(\alpha) \) is the highest power of \( x_i \) occurring in a monomial of \( \alpha \), i.e., it is the degree of \( \alpha \) when interpreting all variables besides \( x_i \) as constants.

### Theorem 7.10 (Parameterized Complexity of \( k \)-Termination)

We define the parameterized decision problem \( k \)-termination as follows: \((\varphi, A \cdot \vec{x}), k) \in \mathcal{L}_k\)-termination iff the loop \((\varphi, A \cdot \vec{x})\) terminates over \( S \) and \( A \) has \( k \) eigenvalues.

For uniform loops, \( k \)-termination is in \( \text{XP} \). Moreover, for such loops, \( k \)-termination over \( \mathbb{R} \) is in \( \text{XP} \) as well.
states this observation on Def. 2.3, where the update matrix \( \vec{q} \), i.e., when inserting Lemma 7.12. Since we just ignore the variables which belong to the eigenvalue zero, we can consider the uniform loop and its normalized closed form.

Definition 7.11 (Hierarchical Expression). Let \( Q_S[\vec{x}]_{\text{lin}} \) denote the set of linear polynomials from \( Q_S[\vec{x}] \), i.e., of degree at most 1. An expression \( h \in \text{NPE}_S[\vec{x}] \) for some ring \( \mathbb{Z} \leq S \leq \mathbb{R}_\lambda \) is a hierarchical expression over the indices \( 1 \leq i_1 < \ldots < i_\nu \leq d \) if there exist \( 1 \leq r \leq \nu \) and \( \lambda \in S_{>0} \) such that

\[
h = \sum_{\lambda \in \text{spec}(A), \lambda' < \lambda} n^{r \lambda'} \cdot \lambda,
\]

where \( \alpha_s \in Q_S[x_{i_1}, \ldots, x_{i_\nu}]_{\text{lin}} \), \( \alpha_s(0, \ldots, 0) = 0 \), and \( \deg_{x_{i_1}}(\alpha_s) = 1 \) for \( r \leq s < \nu \). Here, \( \alpha_s(v_8, \ldots, v_{\nu}) \) abbreviates \( \alpha_s[x_{i_1}/v_8, \ldots, x_{i_\nu}/v_{\nu}] \) for \( v_8, \ldots, v_{\nu} \in \mathbb{R}_\lambda \). We call \( (h) = r \) the offset, \( \nu \) the order, and base \( (h) = \lambda \) the base of \( h \).

Lemma 7.12 states this observation on \( \vec{q}_{\text{norm}} \) formally, where the index \( \text{idx}(\lambda) \) is the sum of the algebraic multiplicities of all smaller eigenvalues than \( \lambda \), i.e.,

\[
\text{idx}(\lambda) = \sum_{\lambda' \in \text{spec}(A), \lambda' < \lambda} \nu(\lambda'),
\]

If \( A \)'s smallest eigenvalue \( \lambda_1 = 0 \), then all entries \( q_{1}, \ldots, q_{\text{idx}(\lambda_2)} \) of \( \vec{q}_{\text{norm}} \) are 0, i.e., when inserting \( \vec{q}_{\text{norm}} \) into the loop condition, the variables \( x_1, \ldots, x_{\text{idx}(\lambda_2)} \) vanish. So from now on we assume that 0 is not an eigenvalue of \( A \). (Note that since we just ignore the variables which belong to the eigenvalue zero, we can also permit uniform loops where the eigenvalue 0 may have a higher geometric multiplicity.)

Lemma 7.12. For all \( \lambda \in \text{spec}(A) \) and \( 1 \leq r < \nu(\lambda) \), the \( (\text{idx}(\lambda) + r) \)-th element of \( \vec{q}_{\text{norm}} \) is a hierarchical expression over the indices \( \text{idx}(\lambda) + 1, \text{idx}(\lambda) + 2, \ldots, \text{idx}(\lambda) + \nu(\lambda) \) with offset \( r \), order \( \nu(\lambda) \), and base \( \lambda \).

Example 7.13. Consider the uniform loop and its normalized closed form \( \vec{q}_{\text{norm}} \) in Fig. 10, where the update matrix \( A \) has \( k = 2 \) eigenvalues \( \lambda_1 = 1 \) of algebraic multiplicity \( \nu_1 = 2 \) and \( \lambda_2 = 2 \) of algebraic multiplicity \( \nu_2 = 3 \), both of which have geometric multiplicity one. Moreover, \( \text{idx}(\lambda_1) = 0 \) and \( \text{idx}(\lambda_2) = 0 + \nu_1 = 2 \). Here, we have \( d = 5 \).

For \( \vec{q}_{\text{norm}} \) in Fig. 10, \( h = q_4 = x_4 \cdot 2^n + \frac{5}{2} \cdot n \cdot 2^n \) is a hierarchical expression over the indices \( i_1 = 3, i_2 = 4, i_3 = 5 \) with offset 2, order 3, and base 2, as \( h = \sum_{s=2}^{3} \alpha_s \cdot n^{s-2} \cdot 2^n \) for \( \alpha_2 = x_4 \in \mathbb{R}_\lambda[x_4, x_5] \) and \( \alpha_3 = \frac{5}{2} \in \mathbb{R}_\lambda[x_5] \).

To describe the form of the whole vector \( \vec{q}_{\text{norm}} \), we now introduce hierarchical partitions. To this end, similar to the concept of solvable loops in Def. 2.3, we consider a partitioning of \( \{1, \ldots, d\} \).

Definition 7.14 (Hierarchical Partition). For \( k \geq 1 \), let \( \nu_1, \ldots, \nu_k \in \mathbb{N} \) form a \( k \)-partition of \( d \), i.e., \( \nu_1 + \ldots + \nu_k = d \) and \( \nu_i > 0 \) for all \( 1 \leq i \leq k \). The blocks associated to the partition \( \nu_1, \ldots, \nu_k \) are \( B_1 = \{1, \ldots, \nu_1\} \), \( B_2 = \{\nu_1 + 1, \ldots, \nu_1 + \nu_2\}, \ldots \) and \( B_k = \{\nu_1 + \ldots + \nu_{k-1} + 1, \ldots, d\} \).

The hierarchical expressions \( h_1, \ldots, h_d \) are a hierarchical \( k \)-partition via \( \nu_1, \ldots, \nu_k \) with bases \( 0 < \lambda_1 < \ldots < \lambda_k \) from \( S \) if for all \( 1 \leq i \leq k \):

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
x_{1} + n & x_{2} \\
x_{2} & x_{3} + 2n & \frac{n}{2} \cdot n \cdot 2^n \cdot 2^n \\
x_{3} & x_{4} \cdot 2^n + \frac{5}{2} \cdot n \cdot 2^n \\
x_{4} & x_{5} \cdot 2^n
\end{pmatrix}
\]
Indeed, when transforming the update of a uniform loop to Jordan normal form, then the normalized closed form is always a hierarchical partition.

**Corollary 7.15** (\(\bar{q}_{\text{norm}}\) is Hierarchical Partition). Let \(A\) be the update matrix of a uniform loop with eigenvalues \(0 < \lambda_1 < \ldots < \lambda_k\) and algebraic multiplicities \(\nu_1, \ldots, \nu_k\), and let \(Q\) be its Jordan normal form where the numbers on the diagonal are weakly monotonically increasing. Then the normalized closed form \(\bar{q}_{\text{norm}}\) of the update \(Q\) is a hierarchical \(k\)-partition via \(\nu_1, \ldots, \nu_k\) and bases \(\lambda_1 < \ldots < \lambda_k\). Here the \(i\)-th block is \(B_i = \{ \text{idx}(\lambda_i) + 1, \ldots, \text{idx}(\lambda_i) + \nu_i \}\).

**Example 7.16.** In Fig. 10, \(\bar{h} = \bar{q}_{\text{norm}}\) is a hierarchical 2-partition via \(\nu_1 = 2\) and \(\nu_2 = 3\), blocks \(B_1 = \{1, 2\}, B_2 = \{3, 4, 5\}\), and \(\lambda_1 = 1 < 2 = \lambda_2\). Moreover, \(\text{off}(h_1) = 1\) and \(\text{off}(h_2) = 2\), while \(\text{off}(h_3) = 1, \text{off}(h_4) = 2\), and \(\text{off}(h_5) = 3\).

### 7.3.3 Interval Conditions

Our decision procedure in Sect. 4 instantiates the variables in the polynomials \(f\) of the loop guard by \(\bar{q}_{\text{norm}} = (h_1, \ldots, h_d)\), resulting in poly-exponential expressions \(p = f(h_1, \ldots, h_d)\). We now prove that in the setting of a hierarchical \(k\)-partition \(\bar{q}_{\text{norm}}\), the atoms in \(\text{red}(p > 0)\) (see Lemma 4.7 and (7)) are equivalent to so-called interval conditions, whose satisfiability is particularly easy to check (Sect. 7.3.4). In Lemma 7.19, Lemma 7.21, and Cor. 7.23, we introduce the different subformulas occurring in \(\text{red}(p > 0)\). Here, it is convenient to define the active variables of polynomials.

**Definition 7.17** (Active Variables). Let \(f = c_0 + \sum_{i=1}^{d} c_i \cdot x_i \in S[\bar{x}]_{\text{lin}}\). We define \(\text{actVar}(f) = \{x_i \mid c_i \neq 0\}\). If \(x_i \in \text{actVar}(f)\), then \(\text{coeff}(f, x_i) = c_i\).

**Example 7.18.** Consider \(f = -x_1 + 3 \cdot x_3 + 4 \in S[x_1, \ldots, x_5]_{\text{lin}}\). Then \(\text{actVar}(f) = \{x_1, x_3\}\), \(\text{coeff}(f, x_1) = -1\), and \(\text{coeff}(f, x_3) = 3\).

From the addends \(a_j \cdot n^{x_j} \cdot b_j^a\) of \(p = f(h_1, \ldots, h_d)\) with \(a_j \in Q_{S}[\bar{x}]_{\text{lin}}\), \(a_j \in \mathbb{N}\), and \(b_j \in S_{>0}\), we again compute the set \(\text{coefs}(p)\) of marked coefficients as in Def. 4.5, which have the form \(a_j^{(b_j, a_j)}\). For any \(b \in S_{>0}\) and \(a \in \mathbb{N}\), we now define a formula \(\text{zero}(b, a)\) which is equivalent to requiring that all addends \(a_j \cdot n^{x_j} \cdot b_j^a\) vanish where \(b_j = b\) and where \(a_j \geq a\).

**Lemma 7.19** (Formulas for Vanishing of Addends). Let \(h_1, \ldots, h_d\) be a hierarchical \(k\)-partition via \(\nu_1, \ldots, \nu_k\) with bases \(0 < \lambda_1 < \ldots < \lambda_k\), and let \(f \in S[\bar{x}]_{\text{lin}}\). For any \(1 \leq i \leq k\), let \(F(i) = \{j \in B_i \mid x_j \in \text{actVar}(f)\}\). Let \(M = -\frac{\text{coeff}(f, x_{\text{min}} F(i)) \cdot c_{\text{min}} F(i)}{\text{off}(i)}, 0\) if \(\lambda_i = 1\) and \(F(i) \neq \emptyset\), where \(c_{\text{min}} F(i)\) is the coefficient of \(x_{\text{min}} F(i)\) in \(h_{\text{min}} F(i)\). For any \(b \in S_{>0}\) and \(a \in \mathbb{N}\), we define \(\text{zero}(b, a)\):

\[\text{zero}(b, a) = \text{true}\]
(a) If \( \lambda_i \neq 1 \) for all \( 1 \leq i \leq k \) or \( \lambda_i = 1 \) for some \( 1 \leq i \leq k \) and \( F(i) = \emptyset \), then \( \text{zero}(1,0) \) is the formula \( f(0,\ldots,0) = 0 \).

(b) If \( \lambda_i = 1 \) for some \( 1 \leq i \leq k \) and \( F(i) \neq \emptyset \), then \( \text{zero}(1,0) \) is the formula
\[
x_{\min} F(i) = M \land \bigwedge_{j \in B_i, \min F(i) < j} (x_j = 0).
\]

(c) If \( \lambda_i = b \) for some \( 1 \leq i \leq k \), \( F(i) \neq \emptyset \), and \( (b, a) \neq (1,0) \), then \( \text{zero}(b,a) = \bigwedge_{j \in B_i, a+\min F(i) \leq j} (x_j = 0) \).

(d) Otherwise, we define \( \text{zero}(b,a) = \text{true} \).

Example 7.20. Consider \( f = -x_1 + 3 \cdot x_3 + 4 \) from Ex. 7.18 and \( \vec{h} = \vec{q}_{\text{norm}} \). We have \( p = f(h_1, \ldots, h_5) = -h_1 + 3 \cdot h_3 + 4 = (-x_1 + 4) - x_2 \cdot n + 3 \cdot x_3 \cdot 2^n + (\frac{3x_4}{4} - \frac{3x_5}{8}) \cdot n \cdot 2^n + \frac{3x_5}{8} \cdot n^2 \cdot 2^n \).

In the notation of Lemma 7.19, since \( B_1 = \{1,2\} \), \( B_2 = \{3,4,5\} \), and \( \text{actVar}(f) = \{x_1,x_3\} \), we have \( F(1) = \{j \in \{1,2\} | x_j \in \{x_1,x_3\}\} = \{1\} \) and \( F(2) = \{j \in \{3,4,5\} | x_j \in \{x_1,x_3\}\} = \{3\} \). Thus, \( \min F(1) = 1 \), \( \text{coeff}(x_{\min} F(1)) = -1 \) and thus \( M = -\frac{\lambda_{x_1}}{\lambda_{x_3}} = 4 \). Hence,
\[
\text{zero}(1,0) = (x_1 = 4) \land (x_2 = 0) \land (x_3 = 0) \land (x_4 = 0) \land (x_5 = 0).
\]
So \( \vec{v} \in \mathbb{R}^5 \) satisfies \( \text{zero}(1,0) \) or \( \text{zero}(2,0) \), respectively, iff all terms with base 1 or 2 vanish in \( f(h_1, \ldots, h_5)[\vec{x}/\vec{v}] \).

After introducing \( \text{zero}(b,a) \), we are now ready to show that the formulas in Lemma 4.7 and Thm. 4.9 have a special form when considering uniform loops. In the following lemmas, corollaries, and definitions, let \( h_1, \ldots, h_d, \nu_1, \ldots, \nu_k, f, F(i), p, \text{coeffs \( (p) \)}, \) and \( \ell \) always be as in Lemma 7.19.

Lemma 7.21. For any \( 1 \leq s_0 \leq \ell \), the formula \( \alpha_{s_0} > 0 \land \bigwedge_{i=s_0+1}^{\ell} (\alpha_i = 0) \) is equivalent to the following formula \( \rho_{f,s_0} \), which can be computed in polynomial time from \( h_1, \ldots, h_d \) and \( f \):

(a) If \( (b_{s_0}, a_{s_0}) = (1,0) \) and either \( \lambda_i \neq 1 \) for all \( 1 \leq i \leq k \) or \( \lambda_{i_0} = 1 \) for some \( 1 \leq i_0 \leq k \) and \( F(i_0) = \emptyset \), then \( \rho_{f,s_0} \) is
\[
f(0,\ldots,0) > 0 \land \bigwedge_{i \in \{1,\ldots,k\}, \lambda_i > 1} \text{zero}(\lambda_i,0).
\]

(b) If \( (b_{s_0}, a_{s_0}) = (1,0) \), \( \lambda_{i_0} = 1 \) for some \( 1 \leq i_0 \leq k \), and \( F(i_0) \neq \emptyset \), then there is a \( ^C C \in \mathbb{Q}^S \) with \( C \neq 0 \) such that \( \rho_{f,s_0} \) is
\[
\text{sign}(C) \cdot x_{\min} F(i_0) + \frac{\text{coeff}(f, x_{\min} F(i_0))}{\text{coeff}(f, x_{\min} F(i_0))} > 0 \land \text{zero}(1,1) \land \bigwedge_{i = i_0 + 1}^{k} \text{zero}(\lambda_i,0).
\]

\(^C C = \text{coeff}(f, x_{\min} F(i_0)) \cdot \text{coeff}(f, x_{\min} F(i_0)) \) for the coefficient \( \text{coeff}(f, x_{\min} F(i_0)) \) of \( x_{\min} F(i_0) \) in \( h_{\min} F(i_0) \).
Lemma 7.21. We use Corollary 7.23 to obtain Corollary 7.25. By Corollary 7.23, Lemma 4.7 with Example 7.26.

Example 7.22. For $f, h, p$ of Ex. 7.20, $\text{coeff}(p)$ is

$$\{\alpha_1^{(1.0)}, \alpha_2^{(1,1)}, \alpha_3^{(2.0)}, \alpha_4^{(2,1)}, \alpha_5^{(2,2)}\}$$

where

$$\alpha_1 = -x_1 + 4, \quad \alpha_2 = -x_2, \quad \alpha_3 = 3 \cdot x_3, \quad \alpha_4 = \frac{3x_1 - 3x_3}{2}, \quad \alpha_5 = \frac{3x_2}{5}.$$ 

Let us compute $\rho_{f,4}$. For $\alpha_4^{(2,1)}$, we have $i_0 = 2$, $\min F(i_0) = 3$, and $a_4 = 1$. Moreover, we have $sg = \text{sign}(\text{coeff}(f, x_3) \cdot c_4) = \text{sign}(3 \cdot \frac{1}{5}) = 1$ and $\text{zero}(2, 2) = \sum_{j \in \{3,4,5\}} 2 + 3 \leq j (x_j = 0) = (x_5 = 0)$ and hence by Lemma 7.21:

$$(\alpha_4 > 0) \land (\alpha_5 = 0) \iff (sg \cdot x_{a_4 + \min F(i_0)} > 0) \land \text{zero}(2, a_4 + 1)$$

$$(x_4 > 0) \land (x_5 = 0)$$

For $\alpha_1^{(1.0)}$, as $\lambda_1 = 1$ and $F(1) = \{1\} \neq \emptyset$, by Lemma 7.21 we use $C = \text{coeff}(f, x_1) \cdot 1 = -1$, $f(0, \ldots, 0) = 4$, and $\text{zero}(1, 1) = \sum_{j \in \{1,2,1+1\}} (x_j = 0) = (x_2 = 0)$ to obtain $\rho_{f,1}$, where $\text{zero}(2, 0) = \sum_{j = 3}^5 (x_j = 0)$ by Ex. 7.20:

$$(\alpha_1 > 0) \land \sum_{i = 1}^5 (x_j = 0) \iff (-x_1 + 4 > 0) \land (x_2 = 0) \land \text{zero}(2, 0)$$

In addition to the formulas $\rho_{f,s}$ for $1 \leq s \leq t$, we also introduce a formula $\rho_{f,0}$ which expresses that all coefficients of $p$ vanish.

Corollary 7.23. $\sum_{s = 1}^t (x_s = 0)$ is equivalent to $\rho_{f,0} : \text{zero}(1, 0) \land \sum_{\lambda \in \{1, \ldots, k\}}, \lambda \neq f \emptyset (\lambda = 0)$

Example 7.24. Reconsider Ex. 7.22. By Corollary 7.23, $\prod_{s = 1}^5 (x_s = 0)$ is equivalent to $\rho_{f,0} = \text{zero}(1, 0) \land \text{zero}(2, 0)$, where $\text{zero}(1, 0) = (x_1 = 4) \land (x_2 = 0)$ and $\text{zero}(2, 0) = (x_3 = 0) \land (x_4 = 0) \land (x_5 = 0)$ by Ex. 7.20.

We can now combine Lemma 7.21 and Corollary 7.23 with Lemma 4.7 to obtain the following result. Here, “ic” stands for interval conditions.

Corollary 7.25. For $p \in \{\geq, >\}$, red$(p \circ 0)$ is equivalent to the formula $ic(p \circ 0)$, where $ic(p > 0) = \bigvee_{s = 1}^t \rho_{f,s}$ and $ic(p \geq 0) = \text{inc}(p > 0) \lor \rho_{f,0}$.

Example 7.26. Reconsider Ex. 7.22 and 7.24. By Corollary 7.25, red$(p > 0)$ is equivalent to $ic(p > 0) = \rho_{f,1} \lor \rho_{f,2} \lor \rho_{f,3} \lor \rho_{f,4} \lor \rho_{f,5} =$

$$(-x_1 + 4 > 0 \land x_2 = 0 \land \text{zero}(2, 0)) \lor (-x_2 > 0 \land \text{zero}(2, 0))$$

$$\lor (x_3 > 0 \land x_4 = 0 \land x_5 = 0) \lor (x_4 > 0 \land x_5 = 0) \lor (x_5 > 0),$$

and red$(p \geq 0)$ is equivalent to $ic(p \geq 0) = \text{inc}(p > 0) \lor (\text{zero}(1, 0) \land \text{zero}(2, 0))$.

---

*More precisely, $sg = \text{sign}(\text{coeff}(f, x_{\min F(i_0)}) \cdot c_{\min F(i_0)}^s)$, where $c_{\min F(i_0)}^s$ is the unique coefficient of $x_{\min F(i_0)}^s$ in $h_{\min F(i_0)}$'s addend of the form $\beta \cdot n^\alpha \cdot a_{i_0}^s$, with $\beta \in S_{x_{\min F(i_0)}^s + a_{i_0}, \ldots, x_{i_0} + a_{i_0}}$.***
The formulas $\rho_{f,s}$ in Lemma 7.21 and Cor. 7.23 are so-called interval conditions.

**Definition 7.27 (Interval Condition).** For $1 \leq i, i' \leq d$, $i \neq i'$, $I \subseteq \{1, \ldots, d\}$, $sg \in \{-1, 1\}$, and $0 \neq c \in Q_S$, an interval condition has the following forms:

(a) $\bigwedge_{j \in I} (x_j = 0)$

(b) $sg \cdot x_i > 0 \land \bigwedge_{j \in I \setminus \{i\}} (x_j = 0)$

(c) $x_{i'} = c \land \bigwedge_{j \in I \setminus \{i', i\}} (x_j = 0)$

(d) $sg \cdot x_i > 0 \land x_{i'} = c \land \bigwedge_{j \in I \setminus \{i, i'\}} (x_j = 0)$

(e) $sg \cdot x_i + c > 0 \land \bigwedge_{j \in I \setminus \{i\}} (x_j = 0)$

**Example 7.28.** The formulas $\rho_{f,4} = (x_4 > 0) \land (x_5 = 0)$ and $\rho_{f,1} = (-x_1 + 4 > 0) \land \bigwedge_{j=2}^5 (x_j = 0)$ from Ex. 7.22 are interval conditions as in Def. 7.27 (b) and (e).

### 7.3.4 Checking Satisfiability of Interval Conditions

We now show that to decide satisfiability of the formulas $ic(p > 0)$, we only have to regard instantiations of the variables with values from $\{0, 1, -1, \ast\}$, where $\ast$ stands for one additional non-zero value. There are only polynomially many such instantiations and the particular value for $\ast$ is later determined by SMT solving. This SMT solving only takes polynomial time, because the resulting SMT problem only contains a single variable. Def. 7.29 instantiates variables accordingly and performs Boolean simplifications as much as possible.

**Definition 7.29 (Evaluation).** Let $\rho$ be a propositional formula built from the connectives $\land$ and $\lor$ over atoms of the form $sg \cdot x + c > 0$ and $x = c$ for $sg \in \{1, -1\}$, $c \in Q_S$, and $x \in \{x_1, \ldots, x_d\}$. Moreover, let $\vec{v} \in \{0, 1, -1, \ast\}^d$. The evaluation of $\rho$ w.r.t. $\vec{v}$ (written $\rho(\vec{v})$) results from $\rho(\vec{v}) = \rho[\vec{x}/\vec{v}]$ by simplifying (in)equations without $\ast$ to true or false, and by simplifying conjunctions and disjunctions with true resp. false. We write $\vec{v} \models \rho$ if $\rho(\vec{v}) \neq \bot$.

For example, if $\rho$ is the formula $(x_1 - \frac{5}{2} > 0) \land (x_2 = 0)$ and $\vec{v} = (\ast, 0)$, then $\rho(\vec{v})$ is $\ast - \frac{5}{2} > 0$. Hence, $\vec{v} \models \rho$. So in general, $\vec{v} \models \rho$ means that $\rho(\vec{v})$ is true or that there could be a value $w$ for $\ast$ such that $\rho[\vec{x}/\vec{v}, \ast/\vec{w}]$ is true.

Now we define candidate assignments $\text{cndAssg}(\rho_{f,s})$ for the formulas $\rho_{f,s}$ in Lemma 7.21 and Cor. 7.23 which contain all $\vec{v} \in \{0, 1, -1, \ast\}^d$ that may satisfy $\rho_{f,s}$ (if a suitable value for $\ast$ is found). However, for each Block $B_i$, at most one variable $x_j$ with $j \in B_i$ may be assigned a non-zero value (i.e., $1, -1$, or $\ast$). Moreover, the value $\ast$ may only be used in the block for the eigenvalue $\lambda_i = 1$.

**Definition 7.30 (Sets of Candidate Assignments).** For all $0 \leq s \leq t$, we define:

$$\text{cndAssg}(\rho_{f,s}) = \{\vec{v} \in \{0, 1, -1, \ast\}^d \mid \vec{v} \models \rho_{f,s},$$

$$\forall \ 1 \leq i \leq k. \text{ there is at most one } j \in B_i \text{ with } v_j \neq 0,$$

$$v_j = \ast \implies j \in B_{i_0} \text{ where } \lambda_{i_0} = 1\}$$
Example 7.31. In Ex. 7.22 and 7.24, for \( \rho_{f,4} = (x_4 > 0) \land (x_5 = 0) \), \( \overline{\forall} h \models \rho_{f,4} \) implies \( v_4 = 1 \) and \( v_5 = 0 \). Here, \( v_4 = * \) is not possible, because 4 does not belong to the block \( B_1 = \{1,2\} \) for the eigenvalue 1. Since at most one value for each block may be non-zero, we have \( v_3 = 0 \). In contrast, \( v_1 \) and \( v_2 \) can be arbitrary, but at most one of them may be non-zero. Hence, we obtain the following for \( \rho_{f,4} \) and for \( \rho_{f,0} = (x_1 = 4) \land \bigwedge_{j=2}^{5} (x_j = 0) \):

\[
\text{cndAssg}(\rho_{f,4}) = \{ [0,1,0], [0,0,1], [0,0,0], [0,1,1], [0,1,0], [1,0,1], [1,0,0], [1,1,0], [1,1,1] \}, \quad \text{cndAssg}(\rho_{f,0}) = \{ [0,0] \}
\]

Lemma 7.32. \(|\text{cndAssg}(\rho_{f,s})| \leq (3 \cdot \max\{\nu_i \mid 1 \leq i \leq k\} + 1)^k\) for all \( 0 \leq s \leq \ell \).

Lemma 7.32 is crucial for our algorithm to decide \( k \)-termination for uniform loops: \(|\text{cndAssg}(\rho_{f,s})| \) is bounded by a polynomial in \( d \) if \( k \) is assumed to be a parameter. This is because the \( \nu_i \) form a \( k \)-partition of \( d \), i.e., \( \nu_i \leq d \) for \( 1 \leq i \leq k \). Hence, computing \( \text{cndAssg}(\rho_{f,s}) \) can be done in polynomial time for fixed \( k \).

Example 7.33. In Ex. 7.31, we have \( \nu_1 = 2 \) and \( \nu_2 = 3 \), and thus, \( k = 2 \) and \( \max\{\nu_i \mid 1 \leq i \leq k\} = 3 \). Here, \(|\text{cndAssg}(\rho_{f,4})| = 7 \leq 100 = 10^2 = (3 \cdot 3 + 1)^2 \).

This example shows that the bound in Lemma 7.32 is coarse, but it suffices for our analysis. We now combine Cor. 7.25 and Def. 7.30 to obtain the sets of candidate assignments for the disjunctions \( \text{ic}(p > 0) \) and \( \text{ic}(p \geq 0) \).

Corollary 7.34. We lift \( \text{cndAssg} \) to inequations by defining

\[
\text{cndAssg}(\text{ic}(p > 0)) = \bigcup_{s=1}^{\ell} \text{cndAssg}(\rho_{f,s})
\]

and

\[
\text{cndAssg}(\text{ic}(p \geq 0)) = \text{cndAssg}(\text{ic}(p > 0)) \cup \text{cndAssg}(\rho_{f,0}).
\]

Then we have

\[
|\text{cndAssg}(\text{ic}(p > 0))| \leq (d + 2) \cdot (3 \cdot \max\{\nu_i \mid 1 \leq i \leq k\} + 1)^k.
\]

For a uniform loop with condition \( \varphi \) and normalized closed form \( \vec{q}_{\text{norm}} = \vec{h} \), let \( \varphi(\vec{h}) \) contain the atoms \( f(\vec{h}) > 0 \), where \( f \in \mathcal{S}[\vec{x}]_{\text{lin}} \). To decide termination, our algorithm computes \( \text{cndAssg}(\text{ic}(f(\vec{h}) > 0)) \) for all these atoms \( f(\vec{h}) > 0 \), and then checks for each of the candidate assignments whether it is a witness for eventual non-termination. We first lift \( \text{ic} \) and \( \text{cndAssg} \) to linear formulas.

Definition 7.35 (\( \text{ic} \) and \( \text{cndAssg} \) for Linear Formulas). Let \( \varphi \) be a linear formula over the atoms \( \{f > 0 \mid f \in \mathcal{S}[\vec{x}]_{\text{lin}}, \triangleright \in \{>, \geq\} \} \) and let \( \vec{h} = (h_1, \ldots, h_d) \) be a hierarchical \( k \)-partition. Then the formula \( \text{ic}(\varphi(\vec{h})) \) results from replacing each atom \( f(\vec{h}) > 0 \) in \( \varphi(\vec{h}) \) by \( \varphi[\vec{x}/\vec{h}] \) by \( \text{ic}(f(\vec{h}) > 0) \). By \( \text{cndAssg}(\text{ic}(\varphi(\vec{h}))) \) we denote the set \( \bigcup_{f(\vec{h}) > 0} \text{atom in } \varphi(\vec{h}) \text{ cndAssg}(\text{ic}(f(\vec{h}) > 0)) \).
To analyze termination of uniform loops, we now present an algorithm to
decide whether for a hierarchical $k$-partition $\vec{h}$ and a linear formula $\varphi$,
\[
\exists \vec{x} \in S^d, n_0 \in \mathbb{N}. \quad \forall n \in \mathbb{N}_{>n_0}. \quad \varphi(\vec{h})
\]  
(see (9))
is valid. Our algorithm calls a method SMT($\psi, V, S$) which checks whether
the linear formula $\psi$ in the variables $V$ is satisfiable. Here, the variables $V$
range over $S \in \{Z, Q, R_A\}$ and the coefficients of the polynomials are from
$Q_\Sigma$. (So for $S = Z$, one can first multiply all inequations in $\psi$ by the main
denominator to result in coefficients from $Z$.) In our case, $V = \{\star\}$ and thus,
$|V| = 1$. With these restrictions, the method SMT has polynomial runtime (see
[2, 40]). More precisely, SMT is called in Alg. 4 to determine whether $\star$ can
be assigned a non-zero value such that $\psi(\vec{v}) \downarrow$ is satisfiable. Here, we have to
assign all occurrences of $\star$ in the formula $\psi(\vec{v}) \downarrow$ the same value.

\begin{algorithm}
\caption{Checking Interval Conditions}
\begin{algorithmic}
    \begin{itemize}
    \item \textbf{Input:} a formula $\varphi$ over the atoms \{ $f \geq 0 \mid f \in [S]\_\text{lin}, \varphi \in \{>, \geq\}$ \},
    a hierarchical $k$-partition $\vec{h} = (h_1, \ldots, h_d)$, and a ring $S \in \{Z, Q, R_A\}$
    \item \textbf{Result:} $\top$ if $\exists \vec{x} \in S^d$. $\psi(\varphi(\vec{h}))$ is valid, $\bot$ otherwise
    \item $\psi \leftarrow \text{ic}(\varphi(\vec{h}))$
    \item \textbf{foreach} $\vec{v} \in \text{cndAssg}(\psi)$ \textbf{do}
        \begin{itemize}
        \item $\psi' \leftarrow \psi(\vec{v}) \downarrow$
        \end{itemize}
    \item \textbf{if SMT($\psi' \wedge \star \neq 0$, $\{\star\}, S$) then return $\top$}
    \item return $\bot$
    \end{itemize}
\end{algorithmic}
\end{algorithm}

Let us discuss the complexity of Alg. 4. The formula $\text{ic}(\varphi(\vec{h}))$ and each
element of $\text{cndAssg}(\text{ic}(\varphi(\vec{h})))$ can be computed in polynomial time. By Cor. 7.34,
$\text{cndAssg}(\text{ic}(\varphi(\vec{h})))$ has at most $|\varphi| \cdot (d + 2) \cdot (3 \cdot \max\{|\nu_i| \mid 1 \leq i \leq k\} + 1)^k$
elements, where $|\varphi|$ is the number of atoms in $\varphi$ and $\nu_i \leq d$ for all $1 \leq i \leq k$.
Thus, when considering $k$ to be a parameter, $\text{cndAssg}(\text{ic}(\varphi(\vec{h})))$ can be
computed in polynomial time. Moreover, evaluating a formula w.r.t. $\vec{v}$ according
to Def. 7.29 is possible in polynomial time, too. Finally, SMT has polynomial runtime
as discussed before. So the runtime of the algorithm is polynomial when regarding $k$
as a parameter. We now prove that Alg. 4 is sound and complete.

**Theorem 7.36.** Alg. 4 returns $\top$ iff $\exists \vec{x} \in S^d$. $\text{ic}(\varphi(\vec{h}))$ is valid.

**Example 7.37.** Consider the uniform loop in Fig. 10 where $\varphi = f > 0 \land f' > 0$
for $f = -x_1 + 3 \cdot x_3 + 4$ and $f' = 2 \cdot x_1 - 5$. Let $\vec{h} = \vec{q}_\text{norm}$ as in Fig. 10 and
let $p = f(\vec{h})$ and $p' = f'(\vec{h}) = (2 \cdot x_1 - 5) + 2 \cdot x_2 \cdot n$. Here, $\psi = \text{ic}(\varphi(\vec{h}))$
= $\text{ic}(p > 0) \land \text{ic}(p' > 0)$, where $\text{ic}(p > 0) = \bigvee_{s=1}^{2} \rho_{f,s}$ is stated in Ex. 7.26. Note
that $\text{coeff}(p') = \{ \alpha_1^{(1,0)}, \alpha_2^{(1,1)} \}$ with $\alpha_1^{(1,0)} = 2 \cdot x_1 - 5$ and $\alpha_2^{(1,1)} = 2 \cdot x_2$. Hence,
$\text{ic}(p' > 0) = \rho_{f',1} \lor \rho_{f',2}$. To compute $\rho_{f',1}$, for $C = \text{coeff}(f', x_1) \cdot c_1 = 2 \cdot 1 = 2$
and $f'(0, 0, \ldots, 0) = -5$, Lemma 7.21 (b) results in $\rho_{f',1} = (x_1 - \frac{5}{2} > 0) \land (x_2 = 0)$. For $\rho_{f',2}$, with $sg = \text{sign}(\text{coeff}(f', x_1) \cdot c_2) = \text{sign}(2 \cdot 1) = 1$, Lemma 7.21
\( (d) \) results in \( \rho_{f,2} = (x_2 > 0) \). Now from ic\((p > 0)\) let us choose the disjunct \( \rho_{f,1} = (-x_1 + 4 > 0) \land \wedge_{j=2}^5 (x_j = 0) \) and from ic\((p' > 0)\) let us choose the disjunct \( \rho_{f,1}' = (x_1 - \frac{5}{2} > 0) \land (x_2 = 0) \). We consider \( \vec{v} = (\ast, 0, 0, 0, 0) \). Then

\[
(\rho_{f,1} \land \rho_{f,1})(\vec{v}) = (-\ast + 4 > 0) \land (\ast - \frac{5}{2} > 0)
\]

is satisfiable with the model \( \ast = 3 \). Hence, this model also satisfies \( \psi(\vec{v}) \land \ast \neq 0 \). Thus, for both \( S \in \{ Q, R_A \} \), Alg. 4 proves validity of \( \exists \vec{f} \in S^d, \ ic(\vec{h}) \) and therefore, non-termination of the uniform loop over \( S \).

So for a uniform loop over \( S \in \{ Q, R_A \} \), non-termination is equivalent to validity of \( \exists \vec{f} \in S^d, \forall n \in N_{>n_0}, \varphi(\vec{g}_{\text{norm}}) \), which in turn is equivalent to a formula only containing interval conditions. This insight reduces the search space for proving validity drastically. Thus, we can now prove Thm. 7.10 for \( S \in \{ Q, R_A, R \} \).

**Proof of Thm. 7.10.** For \( S \in \{ Q, R_A \} \), we first transform the uniform loop such that the update matrix is in Jordan normal form and then compute the normalized closed form as in Eq. (23) in polynomial time. This closed form is a hierarchical partition by Cor. 7.15. By combining Cor. 7.25 and Thm. 7.36, Alg. 4 can decide validity of the formula from Thm. 4.9, i.e., termination of the transformed loop (which is equivalent to termination of the original loop by Cor. 5.16).

As the computation of the equivalent interval conditions in Cor. 7.25 clearly works in polynomial time and we have discussed that Alg. 4 runs in polynomial time when \( k \) is assumed to be a parameter, this proves the statement. Finally, these loops terminate over \( R_A \) iff they terminate over \( R \) by Cor. 4.11.

For Thm. 7.10, it was crucial to transform the loop such that the update matrix is in Jordan normal form. Here we relied on a special closed form for the Jordan normal form, while in Sect. 7.1 we only used the transformation to argue why the closed form is computable in polynomial time. Thus, the transformation from Sect. 5 does not only generalize our results from Sect. 4 to a wider class of loops but it also gives rise to novel results like Thm. 7.10.

The approach in the proof of Thm. 7.10 also works for uniform loops over \( Z \) if the update matrix is already in Jordan normal form. But otherwise, in addition to \( \varphi(\vec{g}_{\text{norm}}) \) we also have an update-invariant and \( \text{Th}_{\exists}(S, R_A) \)-definable subset \( F \) which stems from the transformation into Jordan normal form (see Sect. 5). Thus, to decide termination we have to decide validity of \( \exists \vec{f} \in R_A^d, \forall n \in N_{>n_0}, \varphi(\vec{g}_{\text{norm}}) \land \psi_F \). We will discuss this in the next section.

### 7.3.5 Termination of Uniform Loops Over the Integers

Now we show that deciding termination of uniform loops \( (\varphi, A \cdot \vec{x}) \) over the integers is also in \( XP \). Let \( A \in Z^{d \times d} \) with \( k \) integer eigenvalues each of geometric multiplicity one. Then there is a matrix \( T \in Q^{d \times d} \) such that \( A = T^{-1} \cdot Q \cdot T \) for a matrix \( Q \) in Jordan normal form. However, in general we do not have
$T \cdot T^{-1} \in \mathbb{Z}^{d \times d}$ (see, e.g., [58]). As before, let $\eta(\vec{x}) = T \cdot \vec{x}$ and $\varphi' = \eta^{-1}(\varphi)$. Then termination of $(\varphi, A \cdot \vec{x})$ on $\mathbb{Z}^d$ is equivalent to termination of $(\varphi', Q \cdot \vec{x})$ on $\mathcal{G}(\mathbb{Z}^d) = T \cdot \mathbb{Z}^d$ by Cor. 5.16. Here, $\mathcal{L}_T = T \cdot \mathbb{Z}^d$ is the set of all integer linear combinations of $T$’s columns, i.e., their lattice. In general, we have $\mathcal{L}_T \neq \mathbb{Z}^d$.

Hence, we now call Alg. 4 with the input $(\varphi', \vec{h}, Q)$, where $\vec{h} = \vec{q}_{\text{norm}}$ is the normalized closed form for the update $Q \cdot \vec{x}$. For $\psi = \text{ic}(\varphi'(*)\vec{h})$, we want to find out if $\psi(\vec{v})$ holds for some $\vec{v} \in \mathcal{L}_T$. Such a $\vec{v} \in \mathcal{L}_T$ would witness eventual non-termination of $(\varphi', Q \cdot \vec{x})$, and since $\mathcal{L}_T$ is update-invariant under $Q \cdot \vec{x}$ by Lemma 5.12, this is equivalent to non-termination of $(\varphi', Q \cdot \vec{x})$ on $\mathcal{L}_T$.

We modify Alg. 4 such that it computes all $\vec{v} \in \text{cnfAssg}(\psi)$ where $\psi(\vec{v})_\land \star \neq 0$ is satisfied by some $\vec{v}'$ that results from $\vec{v}$ by instantiating $\star$ with a suitable number from $\mathbb{Q}$. As shown in Cor. 7.34, for a fixed number of eigenvalues $k$, there are only polynomially many such candidate assignments $\vec{v}$. Note that the formula $\varphi'$ is only built from the connectives $\land$ and $\lor$, and $\psi$ results from $\varphi'$ by replacing each atom $f > 0$ by $\text{ic}(f(\vec{h}) > 0)$. Hence, for every such $\vec{v}$ there is a subset $\{\text{ic}(f_1(\vec{h}) > 0), \ldots, \text{ic}(f_r(\vec{h}) > 0)\}$ of these formulas such that $\vec{v}'$ satisfies them all and such that satisfying these formulas is sufficient for satisfying $\psi$. For each $1 \leq r \leq e$, let $\ell_r = |\text{coefs}(f_r(\vec{h}))|$. By Cor. 7.25, $\text{ic}(f_r(\vec{h}) > 0)$ has the form $\bigvee_{s=0}^{\ell_r} \rho_{f_r,s}$ or $\bigwedge_{s=0}^{\ell_r} \rho_{f_r,s}$. So for every $r$ there is at least one $s$ where $\rho_{f_r,s}(\vec{v}')$ is true. But due to the construction of $\rho_{f_r,s}$ in Lemma 7.21 and Cor. 7.23, there is at most one $0 \leq s \leq \ell_r$ where $\rho_{f_r,s}(\vec{v}')$ is true. Thus, for every $1 \leq r \leq e$, there is a unique $0 \leq s_r \leq \ell_r$ where $\rho_{f_r,s_r}(\vec{v}')$ is true.

By Lemma 7.21 and Cor. 7.23, all $\rho_{f_r,s_r}$ are interval conditions. Thus, for each entry $v_j$ of $\vec{v}'$ we can find out whether $x_j = v_j$ is required by some $\rho_{f_r,s_r}$, or whether $v_j = 0$ is just due to setting variables to zero by default, i.e., the formula would still hold when assigning an arbitrary value from $\mathbb{Q}$ to $v_j$. So every $\vec{v}' \in \text{cnfAssg}(\psi)$ gives rise to a certain set of formulas $\{\rho_{f_1,s_1}, \ldots, \rho_{f_r,s_r}\}$, which in turn results in a certain abstract assignment that indicates for each entry of $\vec{v}'$ whether its actual value is necessary to be a model.

**Definition 7.38 (Abstract Assignment).** Let $I$ be the set of all intervals of the forms $[c, c], (-\infty, c), (c, \infty), (-\infty, \infty)$, or $(c, \infty)$ for $c, d \in \mathbb{Q}$ with $c \leq d$. Then an abstract assignment is an element of $I^d$.

For each of the obtained abstract assignments, we now have to check whether it is satisfied by some value from $\mathcal{L}_T$. Let $N \subseteq \{1, \ldots, d\}$ be those indices where $j \in N$ iff the $j$-th component of the abstract assignment is $[0, 0]$, i.e., iff the $j$-th component must be 0 in order to satisfy all $\rho_{f_r,s_r}$. Then we compute a basis of the sublattice $\mathcal{L}_N = \{\vec{w} \in \mathcal{L}_T \mid w_j = 0 \text{ for } j \in N\}$. To this end, we solve the system of linear equations $\bigwedge_{j \in N}(T \cdot \vec{x})_j = 0$ where $\vec{x} \in \mathbb{Z}^d$. Here as usual, $(T \cdot \vec{x})_j$ denotes the $j$-th component of the vector $T \cdot \vec{x}$. This problem can be solved in polynomial time (see, e.g., [22, 59]). Since in general this system contains more variables than equations, the solutions yield a certain linear dependence between the variables. This dependence can then be used to reduce the number of variables in the system, i.e., it gives rise to a basis of $\mathcal{L}_N$, where each basis vector is represented by a $\mathbb{Z}$-linear combination of the columns of $T$. Let $d' \leq d$ be the rank of the sublattice $\mathcal{L}_N$ (i.e., the
number of its basis vectors) and let \( P \in \mathbb{Q}^{d \times d'} \) be the matrix whose columns form the basis of \( \mathcal{L}_N \).

Let \( N' \subseteq \{1, \ldots, d\} \) be those indices where \( j \in N' \) iff the \( j \)-th component of the abstract assignment is neither \([0, 0]\) nor \((-\infty, \infty)\). So the \( j \)-th component must be from a certain interval in order to satisfy all \( \rho_{f_{j,s}} \). To ease notation, we define \( I_j = (-\infty, \infty) \) if \( j \notin N' \) and let \( K = \prod_{j=1}^d I_j \subseteq \mathbb{Q}^d \). Now we have to decide whether there exists an \( X \in \mathbb{Z}^d \) such that \( P \cdot X \in K \).

Let \( B_1, \ldots, B_k \) again be the blocks from the \( k \)-partition \( h_1, \ldots, h_d \). Note that if there is a block \( B_i \) where some \( \rho_{f_{j,s}} \) requires \( x_j \) with \( j \in B_i \) to be non-zero (i.e., \( j \in N' \)), then \( \rho_{f_{j,s}} \) requires all \( x_{j'} \) with \( j' > j \) and \( j' \in B_i \) to be zero (i.e., \( j' \in N' \)). Thus, since \( \bar{v} \) satisfies all formulas \( \rho_{f_{j,s}} \) for \( 1 \leq r \leq e \), for each block \( B_i \) there can be at most one \( r \in B_i \) where some \( \rho_{f_{j,s}} \) requires \( x_j \) to be non-zero. Hence, for each block \( B_i \) there is at most one \( j \in B_i \) where \( j \notin N' \). Note that containment in an interval can be described by at most 2 inequations, where the strict inequations can be turned into weak ones since the variables only range over the integers. Thus, to describe the required containment in the intervals for all \( x_j \) with \( j \notin N' \), we need at most \( 2 \cdot k \) inequations. In other words, the requirement \( P \cdot X \in K \) can be described by \( 2 \cdot k \) linear inequations where the coefficients are from \( \mathbb{Q} \). Since linear integer programming with rational coefficients and a fixed number of constraints is possible in polynomial time (see [40]), this shows that checking whether a candidate assignment \( \bar{v} \) gives rise to a solution in \( \mathcal{L}_T \) can be done in \( XP \). As \( \text{cndAssg}(\psi) \) is also polynomial for fixed \( k \), \( k \)-termination of uniform loops over \( \mathbb{Z} \) is in \( XP \) as well.

### 8 Conclusion and Related Work

In this work, we studied termination of \( \text{twm} \)-loops, i.e., loops where the update \( \bar{x} \leftarrow \bar{u} \) is a triangular system of polynomial equations and the use of non-linearity in \( \bar{u} \) is mildly restricted. We first presented a reduction from termination of \( \text{twm} \)-loops to \( \text{Th}_3(S) \) in Sect. 3 and 4. This implies decidability of termination over \( S \in \{\mathbb{R}_A, \mathbb{R}\} \) and semi-decidability of non-termination over \( S \in \{\mathbb{Z}, \mathbb{Q}\} \).

In addition, we showed how to transform certain non-\( \text{twm} \)-loops into \( \text{twm} \)-form in Sect. 5, and discussed how this generalizes our results to a wider class of loops. We also showed that \( \text{twm} \)-transformability is semi-decidable.

Afterwards, we analyzed the complexity of deciding termination for different subclasses of \( \text{twm} \)-loops. In Sect. 6, we first showed that linearizing \( \text{twm} \)-loops can be done in double exponential time. In Sect. 7, we used our transformation and decision procedure to prove \( \text{Co-NP} \)-completeness (\( \forall \mathbb{R} \)-completeness) of termination of linear (linear-update) loops with rational (real) spectrum, and based on linearization, that deciding termination of arbitrary \( \text{twm} \)-loops over \( \mathbb{R}_A \) or \( \mathbb{R} \) is in \( 3\text{-EXPTIME} \).

Finally, we showed that for the subclass of uniform loops over \( \mathcal{S} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}_A, \mathbb{R}\} \), termination can be decided in polynomial time, if the number of eigenval-
ues of the update matrix is fixed. So here our decision procedure can be used as an efficient technique for termination analysis.

Related Work

In contrast to automated termination analysis (see e.g., [1, 6, 7, 9, 11, 24, 25, 38, 39, 50]), we investigated decidability of termination for certain classes of loops in Sect. 4. As termination is undecidable in general, decidability results can only be obtained for very restricted classes of programs.

Nevertheless, many techniques used in automated tools for termination analysis (e.g., ranking functions [1, 5, 6, 7, 9, 50]) focus on similar classes of loops, since such loops occur as sub-programs in (abstractions of) real programs. Tools based on these techniques have turned out to be very successful, also for larger classes of programs. Thus, these tools could benefit from integrating our (semi-)decision procedures and applying them instead of incomplete techniques for any sub-program that can be transformed into a twins-loop.

Related work on decidability of termination also considers similar (and often more restricted) classes of loops. For linear conjunctive loops, termination over $\mathbb{R}$ [41, 44, 61, 65] and $\mathbb{Q}$ [10] is decidable. Decidability of termination of linear conjunctive loops over $\mathbb{Z}$ was conjectured to be decidable in [61]. After several partial results [8, 19, 48] this conjecture was confirmed recently in [30]. However, [4] shows that for slight generalizations of linear conjunctive loops over $\mathbb{Z}$, where a non-deterministic update or a single piecewise update of a variable are allowed, termination is undecidable. Tiwari [61] uses the special case of our twins-transformation from Sect. 5 where the loop and the automorphism are linear. In contrast to these results, our approach applies to non-linear loops with arbitrary conditions over various rings.

Linearization is another attempt to handle non-linearity, see Sect. 6. While the update of solvable loops can be linearized [47], the condition cannot. Otherwise, one could linearize any loop ($p = 0, \bar{F}$), which terminates over $\mathbb{Z}$ iff $p$ has no integer root. By [30], this would imply decidability of Hilbert’s Tenth Problem.

In the non-linear case, [42] proves decidability of termination for conjunctive loops on $\mathbb{R}^d$ for the case that the condition defines a compact and connected subset of $\mathbb{R}^d$. In [64], decidability of termination of conjunctive linear-update loops on $\mathbb{R}^d$ with the non-zero minimum property is shown, which covers conjunctive linear-update loops with real spectrum. For general conjunctive linear-update loops on $\mathbb{R}^d$ undecidability is conjectured. Moreover, [63] shows that termination of conjunctive linear-update loops where the update matrix has only periodic real eigenvalues is decidable, which also covers conjunctive linear-update loops with real spectrum. Here, a special case of our transformation from Sect. 5 with linear automorphisms is used. In combination with [47], [63, 64] both yield a decision procedure for termination of conjunctive twins-loops over $\mathbb{R}$. Furthermore, [46] proves that termination of (not necessarily conjunctive) linear-update loops is decidable if the condition describes a
compact set. Finally, [66] gives sufficient criteria for (non-)termination of solvable loops and [43] presents sufficient conditions under which termination of non-deterministic non-linear loops on $\mathbb{R}^d$ can be reduced to satisfiability of a semi-algebraic system.

For linear-update loops with real spectrum over $\mathbb{R}$, we prove $\forall \mathbb{R}$-completeness of termination, whereas [63, 64] do not give tight complexity results. The approach from [66] is incomplete, whereas we present a complete reduction from termination to the respective existential fragment of the first-order theory. The work in [43] is orthogonal to ours as it only applies to loops that satisfy certain non-trivial conditions. Moreover, we consider loops with arbitrary conditions over various rings, while [42, 43, 63, 64] only consider conjunctive loops over $\mathbb{R}$ and [46] only considers loops over $\mathbb{R}$ where the condition defines a compact set.

Regarding complexity, [48] proves that termination of conjunctive linear loops over $\mathbb{Z}$ with update $\bar{x} \leftarrow A \cdot \bar{x} + \bar{b}$ is in PSPACE if $|\bar{x}| \leq 4$ resp. in EXPSPACE if $A$ is diagonalizable. Moreover, in [5] it is shown that existence of a linear (lexicographic) ranking function for linear conjunctive loops over $\mathbb{Q}$ or $\mathbb{Z}$ is Co-NP-complete.

Our Co-NP-completeness result is orthogonal to those results as we allow disjunctions in the condition. Moreover, Co-NP-completeness also holds for termination over $\mathbb{Z}$, while [10, 61] only consider termination over $\mathbb{Q}$ resp. $\mathbb{R}$. Additionally, we showed that $k$-termination of uniform loops over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}_A$, and $\mathbb{R}$ is in XP, where the parameter $k$ is the number of eigenvalues. This result is also orthogonal to [10, 61] since we again allow disjunctions in the condition. Furthermore, existence of a linear (lexicographic) ranking function is not necessary for termination of linear loops. We refer to [29] for further discussion on possible extensions of our results to uniform loops over $\mathcal{S}$, where however the eigenvalues are not from $\mathcal{S}$.

Several works exploit the existence of closed forms for solvable (or similar classes of) loops, e.g., to analyze termination on a given input, to infer runtime bounds, or to reason about invariants [28, 31, 34, 35, 47, 55]. While our approach covers solvable loops with real eigenvalues (by Cor. 5.20), it also applies to loops which are not solvable, see Ex. 5.23. Our transformation of Sect. 5 may also be of interest for other techniques for solvable or other subclasses of polynomial loops, as it may be used to extend the applicability of such approaches.

Acknowledgments

We thank Alberto Fiori for help with the example $L_{\text{non-pspace}}$ from (10) and Arno van den Essen for useful discussions.

References

1. Alias C, Darte A, Feautrier P, Gonnord L (2010) Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs.
2. Basu S, Pollack R, Roy MF (2006) Algorithms in Real Algebraic Geometry. Algorithms and Comp. in Math. 10, Springer, doi: 10.1007/3-540-33099-2
3. Basu S, Mishra B (2017) Computational and quantitative real algebraic geometry. In: Goodman JE, O’Rourke J, Tóth CD (eds) Handbook of Discrete and Computational Geometry, 3rd Ed., CRC, pp 969–1002
4. Ben-Amram AM, Genaim S, Masud AN (2012) On the termination of integer loops. ACM Trans Prog Lang Syst 34(4), doi: 10.1145/2400676.2400679
5. Ben-Amram AM, Genaim S (2014) Ranking functions for linear-constraint loops. J ACM 61(4), doi: 10.1145/2629488
6. Ben-Amram AM, Genaim S (2017) On multiphase-linear ranking functions. In: Proc. CAV, LNCS 10427, pp 601–620, doi: 10.1007/978-3-319-63390-9_22
7. Ben-Amram AM, Doménech JJ, Genaim S (2019) Multiphase-linear ranking functions and their relation to recurrent sets. In: Proc. SAS, LNCS 11822, pp 459–480, doi: 10.1007/978-3-030-32304-2_22
8. Bozga M, Iosif R, Kopczynski F (2014) Deciding conditional termination. Log Methods Comput Sci 10(3), doi: 10.2168/LMCS-10(3:8)2014
9. Bradley AR, Manna Z, Sipma HB (2005) Linear ranking with reachability. In: Proc. CAV, LNCS 3576, pp 491–504, doi: 10.1007/11513988_48
10. Braverman M (2006) Termination of integer linear programs. In: Proc. CAV, LNCS 4144, pp 372–385, doi: 10.1007/11817963_34
11. Brockschmidt M, Cook B, Fuhs C (2013) Better termination proving through cooperation. In: Proc. CAV, LNCS 8044, pp 413–429, doi: 10.1007/978-3-642-39799-8_28
12. Cany JF (1988) Some algebraic and geometric computations in PSPACE. In: Proc. STOC, pp 460–467, doi: 10.1145/62212.62257
13. Cohen PJ (1969) Decision procedures for real and p-adic fields. Commun Pure Appl Math 22(2):131–151, doi: 10.1002/cpa.3160220202
14. Dai L, Xia B (2012) Non-termination sets of simple linear loops. In: Proc. ICTAC, LNCS 7521, pp 61–73, doi: 10.1007/978-3-642-32943-2_5
15. Dantzig GB (1949) Programming in a linear structure. Econometrica 17:73–74, Report of the September 9, 1948 meeting in Madison
16. Downey RG, Fellows MR (1999) Parameterized Complexity. Monographs in Computer Science, Springer, doi: 10.1007/978-1-4612-0515-9
17. van den Essen A, Hubbers E (1996) Polynomial maps with strongly nilpotent Jacobian matrix and the Jacobian conjecture. Linear Algebra Its Appl 247:121 – 132, doi: 10.1016/0024-3795(95)00095-X
18. van den Essen A (2000) Polynomial Automorphisms and the Jacobian Conjecture. Springer, doi: 10.1007/978-3-0348-8440-2
19. Frohn F, Giesl J (2019) Termination of triangular integer loops is decidable. In: Proc. CAV, LNCS 11562, pp 269–286, doi: 10.1007/978-3-030-25543-5_24
20. Frohn F, Hark M, Giesl J (2020) Termination of polynomial loops. In: Proc. SAS, LNCS 12389, pp 89–112, doi: 10.1007/978-3-030-65474-0_5
21. Frohn F (2020) A calculus for modular loop acceleration. In: Proc. TACAS, LNCS 12078, pp 58–76, doi:10.1007/978-3-030-45190-5_4
22. Frumkin MA (1977) Polynomial time algorithms in the theory of linear diophantine equations. In: Proc. FCT, LNCS 56, pp 386–392, doi:10.1007/3-540-08442-8_106
23. Giesbrecht M (1995) Nearly optimal algorithms for canonical matrix forms. SIAM J Comput 24(5):948–969, doi:10.1137/S0097539793252687
24. Giesl J, Aschermann C, Brockschmidt M, Emmes F, Frohn F, Fuhs C, Hensel J, Otto C, Plücker M, Schneider-Kamp P, Ströder T, Swiderski S, Thiemann R (2017) Analyzing program termination and complexity automatically with AProVE. J Autom Reason 58(1):3–31, doi:10.1007/s10817-016-9388-y
25. Giesl J, Rubio A, Sternagel C, Waldmann J, Yamada A (2019) The termination and complexity competition. In: Proc. TACAS, LNCS 11429, pp 156–166, doi:10.1007/978-3-030-17502-3_10
26. Gomory R (1960) An algorithm for the mixed integer problem. Tech. Rep. RM-2597, The RAND Corporation, URL https://www.rand.org/pubs/research_memoranda/RM2597.html
27. Graham RL, Knuth DE, Patashnik O (1994) Concrete Mathematics: A Foundation for Computer Science, 2nd Ed. Addison-Wesley
28. Hark M, Frohn F, Giesl J (2020) Polynomial loops: Beyond termination. In: Proc. LPAR, EPiC 73, pp 279–297, doi:10.29007/nxv1
29. Hark M (2021) Towards complete methods for automated complexity and termination analysis of (probabilistic) programs. PhD thesis, RWTH Aachen University, Germany, doi:10.18154/RWTH-2021-06073
30. Hosseini M, Ouaknine J, Worrell J (2019) Termination of linear loops over the integers. In: Proc. ICALP, LIPIcs 132, doi:10.4230/LIPIcs.ICALP.2019.118
31. Hunenberger A, Kovács L (2021) Algebra-based synthesis of loops and their invariants (invited paper). In: Proc. VMCAI, LNCS 12597, pp 17–28, doi:10.1007/978-3-030-67067-2_2
32. Kantorovich LV (1960) Mathematical methods of organizing and planning production. Manage Sci 6(4):366–422, doi:10.1287/mnsc.6.4.366
33. Kauers M, Paule P (2011) The Concrete Tetrahedron – Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates. Springer, doi:10.1007/978-3-7091-0445-3
34. Kincaid Z, Breck J, Cyphert J, Reps TW (2019) Closed forms for numerical loops. Proc ACM Program Lang 3(POPL), doi:10.1145/3290368
35. Kovács L (2008) Reasoning algebraically about p-solvable loops. In: Proc. TACAS, LNCS 4963, pp 249–264, doi:10.1007/978-3-540-78800-3_18
36. Land AH, Doig AG (1960) An automatic method of solving discrete programming problems. Econometrica 28(3):497–520, doi:10.2307/1910129
37. Landau E (1909) Handbuch der Lehre von der Verteilung der Primzahlen. Teubner
38. Larraz D, Oliveras A, Rodríguez-Carbonell E, Rubio A (2013) Proving termination of imperative programs using Max-SMT. In: Proc. FMCAD,
39. Leike J, Heizmann M (2015) Ranking templates for linear loops. Log Methods Comput Sci 11(1), doi:10.2168/LMCS-11(1:16)2015
40. Lenstra Jr HW (1983) Integer programming with a fixed number of variables. Math Oper Res 8(4):538–548, doi:10.1287/moor.8.4.538
41. Li Y (2014) A recursive decision method for termination of linear programs. In: Proc. SNC, pp 97–106, doi:10.1145/2631948.2631966
42. Li Y (2016) Termination of single-path polynomial loop programs. In: Proc. ICTAC, LNCS 9965, pp 33–50, doi:10.1007/978-3-319-67504-3
43. Li Y (2017) Termination of semi-algebraic loop programs. In: Proc. SETTA, LNCS 10606, pp 131–146, doi:10.1007/978-3-319-69483-2
44. Li Y (2017) Witness to non-termination of linear programs. Theor Comput Sci 681:75–100, doi:10.1016/j.tcs.2017.03.036
45. Matijasevič JV (1970) The Diophantineness of enumerable sets. Dokl Akad Nauk SSSR 191:279–282
46. Neumann E, Ouaknine J, Worrell J (2020) On ranking function synthesis and termination for polynomial programs. In: Proc. CONCUR, LIPIcs 171, doi:10.4230/LIPIcs.CONCUR.2020.15
47. de Oliveira S, Bensalem S, Prevosto V (2016) Polynomial invariants by linear algebra. In: Proc. ATVA, LNCS 9938, pp 479–494, doi:10.1007/978-3-319-46520-3
48. Ouaknine J, Pinto JS, Worrell J (2015) On termination of integer linear loops. In: Proc. SODA, pp 957–969, doi:10.1137/1.9781611973730.65
49. Pia AD, Dey SS, Molinaro M (2017) Mixed-integer quadratic programming is in NP. Math Program 162(1-2):225–240, doi:10.1007/s10107-016-1036-0
50. Podelski A, Rybalchenko A (2004) A complete method for the synthesis of linear ranking functions. In: Proc. VMCAI, LNCS 2937, pp 239–251, doi:10.1007/978-3-540-24622-0
51. Renegar J (1992) On the computational complexity and geometry of the first-order theory of the reals, Part I: Introduction. Preliminaries. The geometry of semi-algebraic sets. The decision problem for the existential theory of the reals. J Symb Comput 13(3):255–300, doi:10.1016/S0747-7171(10)80003-3
52. Robinson J (1949) Definability and decision problems in arithmetic. J Symb Log 14(2):98–114, doi:10.2307/2266510
53. Roch JL, Villard G (1996) Fast parallel computation of the Jordan normal form of matrices. Parallel Process Lett 06(02):203-212, doi:10.1142/S0129626496000200
54. Roche DS (2018) What can (and can’t) we do with sparse polynomials? In: Proc. ISSAC, pp 25–30, doi:10.1145/3208767.3209027
55. Rodríguez-Carbonell E, Kapur D (2004) Automatic generation of polynomial loop invariants: Algebraic foundation. In: Proc. ISSAC, pp 266–273, doi:10.1109/ISSAC.2004.1317362
56. Schaefer M (2009) Complexity of some geometric and topological problems. In: Proc. GD, LNCS 5849, pp 334–344, doi:10.1007/978-3-642-11805-0
57. Schaefer M, Stefankovic D (2017) Fixed points, Nash equilibria, and the existential theory of the reals. Theory Comput Syst 60(2):172–193, doi:10.1007/s00224-015-9662-0
58. Sidorov SV (2019) On the similarity of certain integer matrices with single eigenvalue over the ring of integers. Math Notes 105(5):756–762, doi:10.1134/S0001434619050122
59. Storjohann A, Labahn G (1996) Asymptotically fast computation of Hermite normal forms of integer matrices. In: Proc. ISSAC, pp 259–266, doi:10.1145/236869.237083
60. Tarski A (1998) A decision method for elementary algebra and geometry. In: Caviness BF, Johnson JR (eds) Quantifier Elimination and Cylindrical Algebraic Decomposition, Springer, pp 24–84, doi:10.1007/978-3-7091-9459-1, originally appeared in 1951, U California Press, Berkeley and Los Angeles
61. Tiwari A (2004) Termination of linear programs. In: Proc. CAV, LNCS 3114, pp 70–82, doi:10.1007/978-3-540-27813-9_6
62. TPDB (2003 – 2021) Termination Problems Data Base. URL http://termination-portal.org/wiki/TPDB
63. Wu B, Shen L, Bi Z, Zeng Z (2010) Termination of loop programs with polynomial guards. In: Proc. ICCSA, LNCS 6019, pp 482–496, doi:10.1007/978-3-642-12189-0_42
64. Xia B, Zhang Z (2010) Termination of linear programs with nonlinear constraints. J Symb Comput 45(11):1234–1249, doi:10.1016/j.jsc.2010.06.006
65. Xia B, Yang L, Zhan N, Zhang Z (2011) Symbolic decision procedure for termination of linear programs. Formal Aspects Comput 23(2):171–190, doi:10.1007/s00165-009-0144-5
66. Xu M, Li ZB (2013) Symbolic termination analysis of solvable loops. J Symb Comput 50:28–49, doi:10.1016/j.jsc.2012.05.005
Appendix

A Proofs

A.1 Proof of Thm. 3.2

Proof. (a) By weak non-linearity, \( u_i = c_i \cdot x_i + p_i \) with \( x_i \notin V(p_i) \) for all \( 1 \leq i \leq d \). Then

\[
u_i(\vec{u}) = c_i \cdot (x_i + p_i) + p_i(\vec{u}) = c_i^2 \cdot x_i + c_i \cdot p_i + p_i(\vec{u}).\]

Assume \( x_i \notin V(p_i(\vec{u})) \). As \( x_i \notin V(p_i) \) by weak non-linearity, there is an \( x_j \in V(p_i) \) with \( x_j \neq x_i \) and \( x_i \notin V(u_j) \), which implies \( x_j \succ_{\vec{u}} x_i \). But \( x_j \in V(p_i) \) also implies \( x_i \succ_{\vec{u}} x_j \), which violates well-foundedness of \( \succ_{\vec{u}} \), i.e., it contradicts the triangularity of \( (\varphi, \vec{u}) \).

Hence, \( c_i^2 \) is the coefficient of \( x_i \) in \( u_i(\vec{u}) \). Since \( c_i^2 \geq 0 \), \( (\varphi \wedge \varphi(\vec{u}), \vec{u}(\vec{u})) \) is non-negative.

Note that \( x_i \succ_{\vec{u}} x_j \) implies \( x_i \in V(p_i) \) (then we also have \( x_i \succ_{\vec{u}} x_j \)) or it implies that there is an \( x_k \in V(p_i) \) with \( x_k \notin V(u_k) \) (then we have \( x_i \succ_{\vec{u}} x_k \)). So in both cases, \( x_i \succ_{\vec{u}} x_j \). Thus, \( \succ_{\vec{u}} \subseteq \succ \). As \( \succ \) is well founded, this means that \( \succ_{\vec{u}} \) is well founded, too. Hence, \( (\varphi \wedge \varphi(\vec{u}), \vec{u}(\vec{u})) \) is triangular.

(b) Now we prove that \( (\varphi, \vec{u}) \) does not terminate on \( \vec{e} \in S^d \) iff \( (\varphi \wedge \varphi(\vec{u}), \vec{u}(\vec{u})) \) does not terminate on \( \vec{e} \).

\[
\begin{align*}
(\varphi, \vec{u}) \text{ does not terminate on } \vec{e} & \iff \forall n \in N. \varphi(\vec{a}^n(\vec{e})) \\
& \iff \forall n \in N. \varphi(\vec{a}^{2^n - 1}(\vec{e})) \\
& \iff \forall n \in N. \varphi(\vec{a}^{2^n}(\vec{e})) \\
& \iff \forall n \in N. (\varphi \wedge \varphi(\vec{u}))(\vec{u}(\vec{u}))^{2^n}(\vec{e}) \\
& \iff (\varphi \wedge \varphi(\vec{u}), \vec{u}(\vec{u})) \text{ does not terminate on } \vec{e}.
\end{align*}
\]

A.2 Proof of Lemma 4.4

Proof. Recall that for \( f, g : N \to R, f(n) \in o(g(n)) \) means

\[
\forall m > 0. \exists n_0 \in N. \forall n \in N_{>n_0}. |f(n)| < m \cdot |g(n)|.
\]

First consider the case \( b^*_2 > b_1 \). We have \( b^*_2 = b^*_1 \cdot \left(\frac{b_1}{b^*_2}\right)^n \), where \( \frac{b_1}{b^*_2} > 1 \). As \( n^{a_1} \in o\left(\left(\frac{b_1}{b^*_2}\right)^n\right) \), we obtain \( n^{a_1} \cdot b^*_1 \in o(\left(\frac{b_1}{b^*_2}\right)^n \cdot b^*_1) = o(b^*_2) \subseteq o(n^{a_2} \cdot b^*_2) \), i.e., \( n^{a_1} \cdot b^*_1 \in o(n^{a_2} \cdot b^*_2) \).

Now consider the case \( b_2 = b_1 \) and \( a_2 > a_1 \). Then \( n^{a_1} \cdot b^*_1 \in o(n^{a_2} \cdot b^*_2) \) trivially holds.

A.3 Proof of Equation (5)

Proof. If \( p(\vec{e}) = 0 \), then \( k = 0 \) by Def. 4.5 and hence \( o(p(\vec{e})) = o(k \cdot n^a \cdot b^m) = o(0) \). Otherwise, \( p(\vec{e}) \) has the form

\[
k \cdot n^a \cdot b^m + \sum_{i=1}^{m} k_i \cdot n^{a_i} \cdot b_i^m
\]

for \( k \neq 0 \) and \( m \geq 0 \). We have \( k_i \cdot a_i \) \in coefs \( (p(\vec{e})) \) and hence \( (b, a) \succ_{\vec{e}} (b_i, a_i) \) for all \( 1 \leq i \leq m \). Thus, Lemma 4.4 implies \( n^{a_i} \cdot b_i^m \in o(n^a \cdot b^m) \) and we get

\[
o(p(\vec{e})) = o(k \cdot n^a \cdot b^m + \sum_{i=1}^{m} k_i \cdot n^{a_i} \cdot b_i^m) = o(n^a \cdot b^m) = o(k \cdot n^a \cdot b^m).
\]
A.4 Proof of Equation (6)

Proof. If $k = 0$, the claim is trivial, so assume $k \neq 0$, i.e., $p(\bar{c}) = k \cdot b^n \cdot n^a + p'$ for some $p' \in \text{NPE}_S$. By Lemma 4.4 we have

$$p' \in o(k \cdot b^n \cdot n^a)$$

$$\iff \forall m > 0. \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. |p'| < m \cdot |k \cdot b^n \cdot n^a|$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. |p'| < |k \cdot b^n \cdot n^a|.$$ 

Assume $k > 0$. Then

$$\exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. |p'| < |k \cdot b^n \cdot n^a|$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. -p' < |k \cdot b^n \cdot n^a|$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. -p' < k \cdot b^n \cdot n^a$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. 0 < k \cdot b^n \cdot n^a + p'$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. 0 < p(\bar{c})$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. \text{sign}(p(\bar{c})) = \text{sign}(k).$$ 

If $k < 0$, then

$$\exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. |p'| < |k \cdot b^n \cdot n^a|$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. p' < |k \cdot b^n \cdot n^a|$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. p' < -k \cdot b^n \cdot n^a$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. k \cdot b^n \cdot n^a + p' < 0$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. p(\bar{c}) < 0$$

$$\iff \exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. \text{sign}(p(\bar{c})) = \text{sign}(k).$$ 

\[\square\]

A.5 Proof of Lemma 4.7

Proof. We have $p \in \text{NPE}_S[\bar{x}]$, so $p(\bar{c}) \in \text{NPE}_S$ for any $\bar{c} \in \mathbb{S}^d$. Hence,

$$\exists n_0 \in \mathbb{N}. \forall n \in \mathbb{N}_{>n_0}. p(\bar{c}) > 0 \iff \text{unmark}(\text{max}_{> \text{conf}}(\text{coefs}(p(\bar{c})))) > 0 \text{ (by (6)).}$$

Let $\text{coefs}(p) = \{\alpha_1^{(b_1, a_1)}, \ldots, \alpha_\ell^{(b_\ell, a_\ell)}\}$, where $\alpha_i^{(b_i, a_i)} \prec_{\text{conf}} \alpha_j^{(b_j, a_j)}$ for all $1 \leq i < j \leq \ell$. If $p(\bar{c}) = 0$, then $\alpha_1(\bar{c}) = \ldots = \alpha_\ell(\bar{c}) = 0$ and thus $\text{coefs}(p(\bar{c})) = \{0^{(1, 0)}\}$ and $\text{unmark}(\text{max}_{> \text{conf}}(\text{coefs}(p(\bar{c})))) = 0$. Otherwise, there is an $1 \leq i \leq \ell$ with $\text{unmark}(\text{max}_{> \text{conf}}(\text{coefs}(p(\bar{c})))) = \alpha_i(\bar{c}) \neq 0$ and $\alpha_j(\bar{c}) = 0$ for all $i + 1 \leq j \leq \ell$.

So when defining $\text{red}(p > 0)$ and $\text{red}(p \geq 0)$ as in (7), we obviously have

$$\text{unmark}(\text{max}_{> \text{conf}}(\text{coefs}(p(\bar{c})))) > 0 \iff (\text{red}(p > 0))(\bar{c}) \text{ holds.}$$

Hence, (8) is equivalent to

$$\exists \bar{c} \in \mathbb{S}^d. \text{red}(p > 0). \quad (16)$$

The time needed to compute and sort $\text{coefs}(p)$ is polynomial. Furthermore, $\text{red}(p > 0)$ is a disjunction of at most $\ell + 1$ subformulas, where each subformula is a conjunction of at most $\ell$ (in-)equations over $\text{coefs}(p)$. Thus, the time needed to compute $\text{red}(p > 0)$ resp. (16) is polynomial in the size of $p$. 

\[\square\]
A.6 Proof of Thm. 4.9

Proof. We have to prove
\[ (9) \iff \exists \vec{x} \in S^d. \ \text{red}(\xi) \]  
where \text{red}(\xi) results from replacing each atom \( p \lor 0 \) in \( \xi \) by \text{red}(p \lor 0). Since each \text{red}(p \lor 0)

can be computed in polynomial time due to Lemma 4.7, the computation of the formula
\( \neg \exists \vec{x} \in S^d. \ \text{red}(\xi) \) clearly works in polynomial time, too.

To prove (17), we introduce the notion of a fundamental set. Let \( p_1 \lor 0, \ldots, p_k \lor 0 \)

denote the atoms in \( \xi \). We call a subset \( I \subseteq \{1, \ldots, k\} \) fundamental if \( \bigwedge_{i \in I} p_i \lor 0 \Rightarrow \xi \).

Recall that w.l.o.g., we can assume that \( \xi \) does not contain any connectives except \( \land \) and \( \lor \).

Thus, whenever \( \xi \neq \text{false} \), the formula \( \xi \) must have fundamental sets. Clearly, we have
\[ \exists \vec{x} \in S^d. \ \text{red}(\xi) \iff \exists \text{fundamental set } I. \ \exists \vec{x} \in S^d. \ \bigwedge_{i \in I} \text{red}(p_i \lor 0). \]  
Thus, to prove (17), it suffices to show the following:
\[ (9) \iff \exists \text{fundamental set } I. \ \exists \vec{x} \in S^d. \ \bigwedge_{i \in I} \text{red}(p_i \lor 0). \]  

For the “\( \iff \)”-direction, assume there is such a fundamental set and \( \vec{e} \in S^d \), i.e.,
\[ \bigwedge_{i \in I} \text{red}(p_i \lor 0)(\vec{e}) \]
is valid. Then as in the proof of Lemma 4.7, for each \( i \in I \), there is an \( n_i \in \mathbb{N} \) such that
\[ \exists \vec{x} \in S^d. \ \forall n \in \mathbb{N}_{>n_i}. \ p_i \lor 0. \]
As \( I \) is finite, \( n_{\max} = \max\{n_i \mid i \in I\} \) exists. Hence, we get
\[ \exists \vec{x} \in S^d. \ \forall n \in \mathbb{N}_{>n_{\max}}. \ \bigwedge_{i \in I} p_i \lor 0. \]  
Since \( I \) is fundamental, this implies (9).

For the “\( \Rightarrow \)”-direction, assume (9). Then there is an \( \vec{e} \in S^d \) and an \( n_0 \in \mathbb{N} \) such
that for each \( n \in \mathbb{N}_{>n_0} \), there is a fundamental set \( I_n \) such that \( \bigwedge_{i \in I_n} p_i(\vec{e}) \lor 0 \) holds. As
there are only finitely many fundamental sets, there is some fundamental set \( I \) that occurs infinitely often in \( \langle I_n \rangle_{n \in \mathbb{N}_{>n_0}} \). Hence we get
\[ \exists n_0 \in \mathbb{N}. \ \exists \vec{x} \in S^d. \bigwedge_{i \in I} p_i(\vec{e}) \lor 0. \]  
(19)

By definition of poly-exponential expressions, each \( p_i(\vec{e}) \) is weakly monotonic in \( n \) for large enough \( n \). Thus, (19) implies\(^5\)
\[ \exists n_0 \in \mathbb{N}. \ \forall n \in \mathbb{N}_{>n_0}. \ \bigwedge_{i \in I} p_i(\vec{e}) \lor 0, \]
As \( \vec{e} \in S^d \), there is a fundamental set \( I \) such that \( \exists \vec{x} \in S^d. \ \bigwedge_{i \in I} \text{red}(p_i \lor 0) \) holds. \( \square \)

\(^5\) This corresponds to the observation that if the loop condition is a disjunction (and

hence also \( \xi \) is a disjunction of the form \( \xi_1 \lor \xi_2 \)), then non-termination of the original loop
implies non-termination of one of the loops where instead of the disjunction one only takes
one of the disjuncts as the loop guard. The reason is that for every fundamental set \( I \) and
every \( \vec{e} \in S^d \), \( \bigwedge_{i \in I} p_i(\vec{e}) \lor 0 \Rightarrow (\xi_1(\vec{e}) \lor \xi_2(\vec{e})) \) implies \( \bigwedge_{i \in I} p_i(\vec{e}) \lor 0 \Rightarrow \xi_1(\vec{e}) \) or
\( \bigwedge_{i \in I} p_i(\vec{e}) \lor 0 \Rightarrow \xi_2(\vec{e}) \). The restriction to \text{tran}-loops implies that the closed forms are
poly-exponential expressions and hence, that the \( p_i(\vec{e}) \) are weakly monotonic in \( n \) for large
enough \( n \). Therefore, the above argumentation in the proof shows that there is a fundamental
set \( I \) such that \( \bigwedge_{i \in I} p_i(\vec{e}) \lor 0 \) holds for all large enough \( n \) and thus, for some \( j \in \{1, 2\} \),
\( \xi_j(\vec{e}) \) holds for all large enough \( n \) as well. Hence, the loop would also be non-terminating if
one only takes the corresponding disjunct as the loop guard.
A.7 Proof of Thm. 4.10

Proof. By Cor. 3.3, termination of \( \text{tnn}\) -loops is reducible to termination of \( \text{tnn}\) -loops. Given a \( \text{tnn}\) -loop \((\phi, \bar{u})\), we obtain \( \bar{q}_{\text{norm}} \in (\text{NPE}_{\mathbb{S}}[\mathbf{x}])^\beta \) such that \((\phi, \bar{u})\) is (eventually) non-terminating iff (4) holds, where \( \phi \) is a propositional formula over the atoms \( \{n \geq 0, \alpha > 0 \mid \alpha \in \mathbb{S}[\mathbf{x}]\} \). Hence, \( \phi(\bar{q}_{\text{norm}}) \) is a propositional formula over the atoms \( \{p \geq 0 \mid p \in \text{NPE}_{\mathbb{S}}[\mathbf{x}]\} \). Thus, by Thm. 4.9, validity of (4) resp. (9) is reducible to \( \text{Th}_{\mathbb{S}}(\mathbb{S}) \). 

A.8 Proof of Cor. 4.11

Proof. Again, by Cor. 3.3, termination of \( \text{tnn}\) -loops is reducible to termination of \( \text{tnn}\) -loops. By Thm. 4.10, termination of \( \text{tnn}\) -loops is reducible to invalidity of a closed formula \( \chi \in \text{Th}_{\mathbb{S}}(\mathbb{S}) \). If \( \mathbb{S} = \mathbb{R}_\Lambda \), then validity of \( \chi \) is decidable, and if \( \mathbb{S} = \mathbb{Z} \) or \( \mathbb{S} = \mathbb{Q} \), then validity of \( \chi \) is semi-decidable [15, 60]. But \( \chi \) is valid iff the loop is non-terminating. Hence, non-termination is decidable for \( \mathbb{S} = \mathbb{R}_\Lambda \) and semi-decidable if \( \mathbb{S} = \mathbb{Z} \) or \( \mathbb{S} = \mathbb{Q} \). The claim (b) for \( \mathbb{S} = \mathbb{R}_\Lambda \) follows since deciding non-termination is equivalent to deciding termination. Finally, (a) and the claim (b) for \( \mathbb{S} = \mathbb{R} \) follow due to elementary equivalence of \( \mathbb{R}_\Lambda \) and \( \mathbb{R} \).

A.9 Proof of Lemma 4.12

Proof. We have: \( \bar{e} \) witnesses eventual non-termination of \((\phi, \bar{u})\)

\[
\Leftrightarrow \exists n_0 \in \mathbb{N} \forall n \in N_{>n_0}. \ (\phi(\bar{q}_{\text{norm}}))(\bar{e}) \quad (\text{by (4)})
\]

\[
\Leftrightarrow \exists n_0 \in \mathbb{N} \forall n \in N_{>n_0}. \ (\xi(\bar{e}))
\]

\[
\Leftrightarrow \text{red}(\xi(\bar{e})) \quad (\text{as in the proof of Thm. 4.9})
\]

A.10 Proof of Lemma 5.6

Proof. Let \((\phi, \bar{u})\) be a loop. Since \( id_{\mathbb{S}[\mathbf{x}]}^{-1} = id_{\mathbb{S}[\mathbf{x}]} \), we obtain \( Tr_{id_{\mathbb{S}[\mathbf{x}]}}(\phi, \bar{u}) = (\phi', \bar{u}') \) with

\[
\phi' = id_{\mathbb{S}[\mathbf{x}]}^{-1}(\phi) = \phi
\]

\[
\bar{u}' = (id_{\mathbb{S}[\mathbf{x}]}^{-1} \circ \bar{u} \circ id_{\mathbb{S}[\mathbf{x}]})(\bar{x}) = \bar{u}(\bar{x}) = \bar{u}
\]

Now we take \( n_1, n_2 \in \text{Aut}_{\mathbb{S}}(\mathbb{S}[\mathbf{x}]) \). Note that \( (n_1 \circ n_2)^{-1} = n_2^{-1} \circ n_1^{-1} \). Let \( Tr_{n_1 \circ n_2}(\phi, \bar{u}) = (\phi'', \bar{u}'') \), \( Tr_{n_1}(\phi, \bar{u}) = (\phi', \bar{u}') \), and \( Tr_{n_2}(\phi'', \bar{u}'') = (\phi'', \bar{u}'') \). We have

\[
\phi'' = (n_2^{-1} \circ n_1^{-1})(\phi)
\]

\[
\phi''' = n_2^{-1}(\phi')
\]

\[
\phi'''' = n_2^{-1}(\phi'')
\]

Moreover, we have \( \bar{u}' = (n_2^{-1} \circ n_1^{-1} \circ \bar{u} \circ n_1 \circ n_2)(\bar{x}) \)

\[
= (n_2(\bar{x}))(n_1(\bar{x}))(\bar{u})(n_1^{-1}(\bar{x}))(n_2^{-1}(\bar{x}))
\]

\[
= (n_1(\bar{x}))(\bar{u})(n_1^{-1}(\bar{x}))
\]

\[
= (n_2^{-1}(\bar{u}) \circ n_2)(\bar{x})
\]

\[
= (n_2(\bar{x}))(n_1(\bar{x}))(\bar{u})(n_1^{-1}(\bar{x}))(n_2^{-1}(\bar{x}))
\]

\[
= \bar{u}'
\]
A.11 Proof of Lemma 5.7

Proof. Let $\bar{e} \in S^d$ and $n \in \mathbb{N}$. Then

\[
\begin{align*}
\varphi'((\bar{u}')^n((\eta(\bar{x}))(\bar{e}))) \\
&= \eta^{-1}(\varphi) ((\bar{u}')^n((\eta(\bar{x}))(\bar{e}))) \\
&= \varphi[\bar{x}/\eta^{-1}(\bar{x})] [\bar{x}/\bar{u}'^n][\bar{x}/\eta(\bar{x})] [\bar{x}/\bar{e}] \\
&= \varphi[\bar{x}/\eta^{-1}(\bar{x})] [\bar{x}/\bar{u}] [\bar{x}/\eta^{-1}(\bar{x})] [\bar{x}/\bar{e}] \\
&= \varphi[\bar{x}/\bar{u}] [\bar{x}/\bar{e}] \\
&= \varphi(\bar{u}^n(\bar{e})) \\
\end{align*}
\]

Hence, $\bar{e}$ is indeed a bijection with inverse mapping $\eta^{-1}$.

A.12 Proof of Thm. 5.10

Proof. In Cor. 5.8, we have seen that if $\bar{e}$ is a witness for (eventual) non-termination of $(\varphi, \bar{u})$, then $\bar{e}$ witnesses (eventual) non-termination of $Tr_\eta(\varphi, \bar{u})$. Now let $\bar{e}'$ be a witness for (eventual) non-termination of $Tr_\eta(\varphi, \bar{u})$. Then by Cor. 5.8, $\eta^{-1}(\bar{e}')$ witnesses (eventual) non-termination of $Tr_{\eta^{-1}}(Tr_\eta(\varphi, \bar{u}))$ \textbf{Lemma 5.6} $Tr_{\eta^{-1}}(\varphi, \bar{u}) = (\varphi, \bar{u})$. Hence, $\bar{e}$ maps witnesses for (eventual) non-termination of $(\varphi, \bar{u})$ to witnesses for (eventual) non-termination of $Tr_\eta(\varphi, \bar{u})$ and $\eta^{-1}$ maps witnesses for (eventual) non-termination of $Tr_\eta(\varphi, \bar{u})$ to witnesses for $(\varphi, \bar{u})$. These two mappings are inverse to each other: For $\bar{e}' \in S^d$ we have

\[
\begin{align*}
\bar{e}' &= \eta^{-1}(\eta^{-1}(\bar{e}')) \quad \text{(by definition of } \eta^{-1}) \\
\bar{e}' &= (\eta(\bar{x}))(\eta^{-1}(\bar{x}))(\bar{e}')) \quad \text{(by definition of } \bar{e}) \\
\bar{e}' &= \eta^{-1}(\bar{e}) \quad \text{(by definition of } \eta^{-1}) \\
\bar{e}' &= \bar{e}. \\
\end{align*}
\]

Hence, $\bar{e}$ is indeed a bijection with inverse mapping $\eta^{-1}$.

A.13 Proof of Lemma 5.12

Proof. Let $\bar{e}' \in \bar{e}(F)$. Then $\bar{e}' = \bar{e}(\bar{e})$ for some $\bar{e} \in F$. As $F$ is $\bar{u}$-invariant, we have $\bar{u}(\bar{e}) \in F$. We obtain

\[
\begin{align*}
\bar{u}'(\bar{e}') &= (\eta(\bar{x}))(\bar{u})(\eta^{-1}(\bar{x}))(\bar{e}') \\
&= (\eta(\bar{x}))(\bar{u})(\eta^{-1}(\bar{x}))(\eta(\bar{x}))(\bar{e}) \\
&= (\eta(\bar{x}))(\bar{u})(\bar{e}) \\
&= \bar{u}(\bar{e}) \in \bar{e}(F). \\
\end{align*}
\]

A.14 Proof of Lemma 5.13

Proof. Let $F$ be defined by $\psi_F \in Th_\exists(S, R_A)$. Consider the following formula $\psi \in Th_\exists(S, R_A)$:

\[
\exists \bar{y} \in R^A, \psi_F(\bar{y}) \land \bar{x} = (\eta(\bar{x}))(\bar{y}).
\]

Then $\psi(\bar{e})$ holds for $\bar{e} \in R^A$ iff $\bar{e} = \bar{u}(\bar{e})$ for some $\bar{u} \in R_A$ with $\psi_F(\bar{u})$, i.e., with $\bar{u} \in F$. □
A.15 Proof of Thm. 5.15

Proof. As \((\varphi, \bar{u})\) is solvable, there is a partitioning \(J = \{J_1, \ldots, J_k\}\) as in Def. 2.3, i.e., \(\{1, \ldots, d\} = \bigcup_{i=1}^{k} J_i\) and \(\bar{u}^{-1}_{J_i} = A_i \bar{x}_{J_i} + \bar{p}_i\) for all \(1 \leq i \leq k\), where \(\bar{p}_i \in S[\bar{x}_{J_{i+1}}, \ldots, \bar{x}_{J_k}]^d\).

W.l.o.g. \(\bar{x}\) is ordered according to \(J\), i.e., if \(x_{i_1} \in J_{j_1}\) and \(x_{i_2} \in J_{j_2}\) for \(j_1 < j_2\), then \(i_1 < i_2\).

For each \(A_i\), let \(Q_i = T_i \cdot A_i \cdot T_2^{-1}\) be its Jordan normal form, where \(T_i\) is the corresponding transformation matrix. Since \(\bar{A}_i\) has only real eigenvectors, this means that the entries of \(Q_i, T_i,\) and \(T_2^{-1}\) are real algebraic numbers. Let \(\eta\) be the endomorphism defined by \(\eta(\bar{x}_{J_i}) = T_i \cdot \bar{x}_{J_i}\). This means that \(\eta\) is induced by the block diagonal matrix \(\operatorname{Diag}(T_1, T_2, \ldots, T_k)\).

Then \(\eta\) is an automorphism and its inverse satisfies \(\eta^{-1}(\bar{x}_{J_i}) = T_i^{-1} \cdot \bar{x}_{J_i}\). Furthermore, the degree of \(\eta\) is obviously 1. Moreover, \(\eta\) and \(\eta^{-1}\) are compatible with the partition, i.e., the images of the variables in \(\bar{x}_{J_i}\) under \(\eta\) and \(\eta^{-1}\) are polynomials only using the variables \(\bar{x}_{J_i}\).

For each \(1 \leq i \leq k\) we have:

\[
\begin{align*}
(\eta^{-1} \circ \bar{u} \circ \eta)(\bar{x}_{J_i}) \\
= \eta(\bar{x}_{J_i})[\bar{x}/\bar{u}][\bar{x}/\eta^{-1}(\bar{x})] \\
= (T_i \cdot \bar{x}_{J_i})[\bar{x}/\eta^{-1}(\bar{x})] \\
= (T_i \cdot \bar{u}_{J_i})[\bar{x}/\eta^{-1}(\bar{x})] \\
= (T_i \cdot (A_i \cdot \bar{x}_{J_i} + \bar{p}_i))[\bar{x}/\eta^{-1}(\bar{x})] \\
= (T_i \cdot A_i \cdot \bar{x}_{J_i} + T_i \cdot \bar{p}_i)[\bar{x}/\eta^{-1}(\bar{x})] \\
= (T_i \cdot A_i \cdot \bar{x}_{J_i} + \bar{x}_{J_i}) + (T_i \cdot \bar{p}_i)[\bar{x}/\eta^{-1}(\bar{x})] \\
= (Q_i \cdot \bar{x}_{J_i}) + (T_i \cdot \bar{p}_i)[\bar{x}/\eta^{-1}(\bar{x})]
\end{align*}
\]

We have \(T_i \cdot \bar{p}_i \in S[\bar{x}_{J_{i+1}}, \ldots, \bar{x}_{J_k}]^d\). Therefore, \((T_i \cdot \bar{p}_i)[\bar{x}/\eta^{-1}(\bar{x})] \in S[\bar{x}_{J_{i+1}}, \ldots, \bar{x}_{J_k}]^d\) as well, since \(\eta^{-1}\) is compatible with the partitioning. This implies that \(\operatorname{Tr}_\eta(\varphi, \bar{u})\) is weakly non-linear. As we assumed that \(\bar{x}\) is ordered w.r.t. the partitioning and each \(Q_i\) is triangular, \(\operatorname{Tr}_\eta(\varphi, \bar{u})\) is triangular, too. Thus, \(\operatorname{Tr}_\eta(\varphi, \bar{u})\) is in \(\text{twn}\)-form. \(\square\)

A.16 Proof of Lemma 5.18

Proof. Let \(\delta = \deg(\eta)\). For any \(k \in \mathbb{N}\), there is only a finite number of monomials over \(\bar{x}\) of degree \(k\). (The number of monomials of exactly degree \(k\) is \(\binom{d+k-1}{k}\), see the proof of Thm. 6.11 (b).) Hence, for any \(1 \leq i \leq d\) we can construct the following term that stands for \(\eta^{-1}(x_i)\):

\[
\sum_{m} a_{i,m} \cdot m
\]

Here, the monomials \(m\) contain the variables \(\bar{x}\) and the \(a_{i,m}\) are variables that stand for the unknown coefficients of the polynomial \(\eta^{-1}(x_i)\). Hence, for any \(1 \leq i \leq d\) we now build a formula \(\rho_{\bar{x},i}\) which stands for the requirement

\[
\eta^{-1}(x_i) = \left(\eta^{-1}(x_i) \cdot \eta(x_i)\right) = x_i \quad \text{(i.e., that \(\eta^{-1}\) is a right inverse of \(\eta\))}
\]

\[
\rho_{\bar{x},i} : \sum_{m} a_{i,m} \cdot m = x_i
\]

Similarly, for any \(1 \leq i \leq d\) we construct a formula \(\rho_{\bar{u},i}\) which stands for the requirement

\[
\eta^{-1}(\bar{u})(x_i) = \left(\eta(\bar{u}) \cdot \eta^{-1}(x_i)\right) = x_i \quad \text{(i.e., that \(\eta^{-1}\) is a left inverse of \(\eta\))}
\]

\[
\rho_{\bar{u},i} : \eta(x_i) \left(\sum_{m} a_{i,m} \cdot m\right) = x_i
\]

Thus, the formula

\[
\forall \bar{x} \in \mathbb{R}^d_{\mathbb{A}} : \bigwedge_{i=1}^{d} \rho_{\bar{x},i} \land \bigwedge_{i=1}^{d} \rho_{\bar{u},i}
\]

is valid iff \(\eta\) has an inverse of degree at most \(d^{d-1}\). By Thm. 5.17, this is equivalent to the question whether \(\eta\) has an inverse, i.e., whether \(\eta\) is an automorphism. Unfortunately, \(\bigwedge_{i=1}^{d} \rho_{\bar{x},i} \land \bigwedge_{i=1}^{d} \rho_{\bar{u},i}\) has to hold for all \(\bar{x} \in \mathbb{R}^d_{\mathbb{A}}\). So, we can reduce this formula to a system of equations: one simply has to check whether there is
an instantiation of the unknown coefficients \(a_{i,m}\) such that all monomials in \(\rho_{r,i}\) and \(\rho_{t,i}\) except \(x_i\) get the coefficient 0 and the monomial \(x_i\) gets the coefficient 1. When building the conjunction of these equations and existentially quantifying the unknown coefficients \(a_{i,m}\), one indeed obtains a formula from \(\text{Th}_3(\mathbb{R}_a)\).

**A.17 Proof of Thm. 5.19**

**Proof.** For every \(1 \leq i \leq d\), let

\[ \eta(x_i) = \sum_{m} \text{is a monomial of (at most) degree } \delta \cdot b_{i,m} \cdot m, \]

where the \(b_{i,m}\) are variables that stand for unknown coefficients. By Lemma 5.18 there is a \(\text{Th}_3(\mathbb{R}_a)\)-formula that contains both \(b_{i,m}\) and the variables \(a_{i,m}\) (for the coefficients of \(\eta^{-1}\)) which expresses that \(\eta\) is an automorphism.

Furthermore, using these coefficients we can construct a formula from \(\text{Th}_3(\mathbb{R}_a)\) which expresses that the update \(\vec{u}' = (u'_1, \ldots, u'_d) = (\eta^{-1} \circ \vec{u} \circ \eta)(\vec{e})\) is turn: We have \(\text{deg}(\vec{u}') = \text{deg}(\eta^{-1} \circ \vec{u} \circ \eta)(\vec{e}) \leq \text{deg}(\eta^{-1}) \cdot \text{deg}(\vec{u}) \cdot \text{deg}(\eta) \leq \delta^{d-1} \cdot \text{deg}(\vec{u}) \cdot \delta\). So there is a bound on the degree of the polynomials in the transformed loop \(\text{Tr}_\eta(\vec{u}, \vec{u}')\). For every \(1 \leq i \leq d\), let

\[ u'_i = \sum_{m} \text{is a monomial of (at most) degree } \delta^{d-1} \cdot \text{deg}(\vec{u}) - c_{i,m} \cdot m, \]

where the variables \(c_{i,m}\) stand again for unknown coefficients. Now we can build a \(\text{Th}_3(\mathbb{R}_a)\)-formula which is valid iff \(\vec{u}'\) is in turn-form by requiring that certain coefficients \(c_{i,m}\) are zero. Moreover, we can construct a formula from \(\text{Th}_3(\mathbb{R}_a)\)-formula which is valid iff \(\vec{u}' = (\eta^{-1} \circ \vec{u} \circ \eta)(\vec{e})\).

**A.18 Proof of Cor. 6.3**

**Proof.** The proof is by induction on \(n\). The induction base \(n = 0\) is trivial. In the induction step \(n > 0\) we obtain

\[ \vec{u}([u^{n+1}]) = \vec{u}([u^n]([\vec{u}(\vec{e})])) \]

\[ = (\vec{u}'^n)(\vec{u}(\vec{e})) \quad \text{(by the induction hypothesis for } \vec{u}(\vec{e}) \in \mathcal{S}^d) \]

\[ = (\vec{u}'^n)(\vec{u}(\vec{e})) \quad \text{(by Def. 6.1)} \]

**A.19 Proof of Lemma 6.6**

**Proof.** (a) If there is no \(\vec{e} \in \mathcal{S}^d\) such that \(\vec{e}' = \vec{u}(\vec{e})\), then \(\varphi'(\vec{y}/\vec{e}')\) is false by Def. 6.4.

(b) For any \(n \in \mathbb{N}\), we have

\[ \varphi'[\vec{y}/(\vec{u}'^n)](\vec{u}(\vec{e})) \]

\[ \iff \varphi'[\vec{y}/\vec{u}^n(\vec{e})] \quad \text{(by Cor. 6.3)} \]

\[ \iff \varphi'[\vec{y}/\vec{u}^n(\vec{e})] \land \mathcal{A}_{\varphi'}^{d+1}(w_1([\vec{y}/\vec{u}^n(\vec{e})]) - w_1([\vec{y}/\vec{u}^n(\vec{e})]) = 0) \quad \text{(by Def. 6.4)} \]

\[ \iff \varphi'[\vec{y}/\vec{u}^n(\vec{e})] \land \mathcal{A}_{\varphi'}^{d+1}(w_1(\vec{u}^n(\vec{e})) - w_1(\vec{u}^n(\vec{e})) = 0) \]

\[ \iff \varphi'[\vec{y}/\vec{u}^n(\vec{e})] \land \mathcal{A}_{\varphi'}^{d+1}(0 = 0) \]

\[ \iff \varphi'[\vec{y}/\vec{u}^n(\vec{e})]. \]

Hence, \((\varphi, \vec{u})\) terminates on \(\vec{e} \in \mathcal{S}^d\) iff \((\varphi', \vec{u}')\) terminates on \(\vec{u}(\vec{e})\).
A.20 Proof of Lemma 6.9

Proof. The claim (a) is obvious. The claim (b) is proved by induction on \( i \). In the induction base, let \( i = d \). Since \( x_d \) is minimal w.r.t. \( \succ \), we have \( \text{depdeg}_{\vec{u}}(x_d) = 1 \leq \text{deg}^0 = \text{deg}^{d-d} \).

In the induction step \( i < d \), the claim is obviously true if \( p_i = 0 \). Otherwise, we obtain:

\[
\begin{align*}
\text{depdeg}_{\vec{u}}(x_i) &= \max\{1, \text{depdeg}_{\vec{u}}(p_i)\} \\
&= \max\{1 \cup \{\text{depdeg}_{\vec{u}}(m) \mid m \text{ is a monomial in } p_i\}\} \\
&= \max\{1 \cup \{\sum_{j=1}^d \sum_{i=1}^{\text{deg}^{d-i-1}} z_j \cdot \text{depdeg}_{\vec{u}}(x_i) \mid \text{occurs in } p_i\}\} \\
&\leq \max\{1 \cup \{\sum_{j=1}^d \sum_{i=1}^{\text{deg}^{d-i-1}} z_j \cdot \text{deg}^{d-i} \mid \text{occurs in } p_i\}\} \quad \text{(by induction hypothesis)} \\
&\leq \max\{1 \cup \{\text{depdeg}^{d-i-1} \cup \sum_{j=1}^d \sum_{i=1}^{\text{deg}^{d-i-1}} z_j \mid \text{occurs in } p_i\}\} \\
&= \text{deg}^{d-1}
\end{align*}
\]

The claim in (c) immediately follows from (b).

A.21 Proof of Thm. 6.11

Proof. Let \( z_1, \ldots, z_d \in \mathbb{N} \). We first show that for all monomials \( m \) in \( u_1^{z_1} \cdots u_d^{z_d} \) we have

\[
\text{depdeg}(m) \leq \text{depdeg}(x_1^{z_1} \cdots x_d^{z_d}). \tag{21}
\]

To prove (21), note that \( m \) must have the form \( m_{i,j} \cdot \ldots \cdot m_{i,z_i} \cdots m_{d,j} \cdots m_{d,z_d} \) where the monomial \( m_{i,j} \) occurs in \( u_i \) for all \( 1 \leq i \leq d \) and \( 1 \leq j \leq z_i \). Therefore, we have \( \text{depdeg}(m_{i,j}) \leq \text{depdeg}(x_i) \). This is clear for \( m_{i,j} = x_i \) and for \( m_{i,j} \neq x_i \) it follows directly from the definition of the dependency degree in Def. 6.8. Hence, we can now prove (21):

\[
\begin{align*}
\text{depdeg}(m) &= \text{depdeg}(m_{1,1}) + \ldots + \text{depdeg}(m_{d,z_d}) \\
&\leq z_1 \cdot \text{depdeg}(x_1) + \ldots + z_d \cdot \text{depdeg}(x_d) \\
&= \text{depdeg}(x_1^{z_1} \cdots x_d^{z_d}).
\end{align*}
\]

Now assume that (a) were not true. Then consider the first execution of Line 4 where we compute an update \( u_m \) for a monomial \( m \) with \( \text{depdeg}_{\vec{u}}(m) > \text{mdepdeg} \). The monomial \( m \) resulted from taking a monomial \( x_1^{z_1} \cdots x_d^{z_d} \) from \( \vec{e} \) and constructing \( u_1^{z_1} \cdots u_d^{z_d} \), where \( m \) occurs in \( u_1^{z_1} \cdots u_d^{z_d} \). Since \( m \) is the first monomial whose dependency degree is greater than \( \text{mdepdeg} \), we have \( \text{depdeg}_{\vec{u}}(x_1^{z_1} \cdots x_d^{z_d}) \leq \text{mdepdeg} \). But by (21), this implies \( \text{depdeg}_{\vec{u}}(m) \leq \text{depdeg}_{\vec{u}}(x_1^{z_1} \cdots x_d^{z_d}) \leq \text{mdepdeg} \), which contradicts our assumption.

For (b), since we do not build any update \( u_m \) for the constant monomial \( m_0 \) (a) implies that the number of non-constant monomials \( m \) over the variables \( \vec{x} \) with \( \text{depdeg}(m) \leq \text{mdepdeg} \) is an upper bound on the number of executions of the while-loop. As we have \( \text{deg}(m) \leq \text{depdeg}(m) \) for any monomial by Lemma 6.9 (a), this number is bounded by the number of monomials over the variables \( \vec{x} \) with a degree between 1 and \( \text{mdepdeg} \).

The number of monomials over \( d \) variables with the exact degree \( \text{mdepdeg} \) (this is the number of so-called weak compositions of \( \text{mdepdeg} \) into \( d \) parts) and the number of monomials over \( d \) variables with a degree between 1 and \( \text{mdepdeg} \) is \((\text{mdepdeg}-1)_d + (\text{mdepdeg}-2)_d + \ldots + (\text{mdepdeg}-d)_d = (d+1) \text{mdepdeg} - 1 - \frac{d(d+1)}{2} \).

Termination of Alg. 3 follows from (b) since the while-loop is only executed finitely often.
A.22 Proof of Thm. 6.12

Proof. The statement in (a) is obvious from Alg. 3 and Thm. 6.11, the claim in (b) follows from Lemma 6.6, and the linearity of the update in (c) is again obvious from Alg. 3.

It remains to show that \((\varphi', \vec{u}')\) is triangular and non-negative (i.e., it is weakly non-linear and the coefficient of the monomial \(y_m\) in \(u'_m\) is always non-negative).

For triangularity, we again assume that \(x_i \succ \varphi y_j\) implies \(i > j\) for all \(1 \leq i, j \leq d\). Then we show that \(y_{m'} \succ \varphi y_m\) implies \(m' \succ m\), where \(\succ\) is the lexicographic ordering on monomials. Thus, if \(m' = \vec{z}'\vec{x}'\) and \(m = \vec{z}\vec{x}\), then \(m' \succ m\) holds iff \(\vec{z}' \succ_{\text{lex}} \vec{z}\). Since \(\succ\) is well founded, this implies the well-foundedness of \(\succ_{\varphi}\), i.e., the loop \((\varphi', \vec{u}')\) is triangular.

So let \(m' \neq m\) and \(y_m\) occur in \(u'_m\). If \(m' = x_1^{i_1} \cdots x_d^{i_d}\), then this means that \(m\) occurs in \(u_1^{j_1} \cdots u_d^{j_d}\). Thus, \(m = m_1 \cdots m_j \cdots m_{d,1} \cdots m_{d,1} \cdots m_1 \cdots m_d \cdots m_d\) where \(m_j\) occurs in \(u_i\), for all \(1 \leq i \leq d\) and \(1 \leq j \leq z_i\). Hence, we have \(m_1 = x_1\) or \(m_d\) only contains variables \(x_j\) with \(x_i \succ \varphi x_j\). Thus, \(x_j \geq x_{j+1}\), where \(\geq\) is the reflexive closure of \(\succ\). Hence, this implies \(m' = x_1^{i_1} \cdots x_d^{i_d} \geq m_1 \cdots m_1 \cdots m_{d,1} \cdots m_{d,1} \cdots m_1 \cdots m_d \cdots m_d\). Since \(m' \neq m\), we have \(m' \succ m\). As \(\succ_{\varphi}\) is the transitive closure of \(\{y_{m'}, y_m\} \mid m\) occurs in \(u'_m\}\), the claim follows.

For non-negativity, note that \(u'_1 \times \cdots \times u'_d\) results from \(u_1^{j_1} \cdots u_d^{j_d}\), where \(u_i = c_i \cdot x_i + p_i\), for all \(1 \leq i \leq d\), and where \(c_i \geq 0\) since \((\varphi, \vec{u})\) is a \(m\)-n-loop. Hence, \(y_{x_1^{j_1} \cdots x_d^{j_d}}\) occurs in \(u'_1 \times \cdots \times u'_d\) in the addend \(c_1 \cdots c_d \cdot y_{x_1^{j_1} \cdots x_d^{j_d}}\). Since this is a linear monomial and since \(c_1 \cdots c_d \geq 0\), this implies non-negativity of \((\varphi', \vec{u}')\).

A.23 Proof of Equation (15)

Proof. We have

\[
\begin{align*}
&(d + \deg d^{-1})^d \\
&= \sum_{i=0}^{d} \binom{d}{i} \cdot \left(\deg d^{-1}\right)^i \cdot d^{d-i} \\
&\leq \sum_{i=0}^{d} \binom{d}{i} \cdot \max \left\{ \deg d^{-1}, d \right\}^d \\
&\leq \sum_{i=0}^{d} \binom{d}{i} \cdot \left( \max \left( \deg d^{-1}, d \right) \right)^d \\
&= 2^d \cdot \left( \max \left( \deg d^{-1}, d \right) \right)^d \\
&= 2^{d+b(\deg) (d-1)} + 2^{d+b(\deg) (d-1)} - 2^{d+b(\deg) (d-1)} - 2^{d+b(\deg) (d-1)} \\
&\leq 2 \cdot 2^{d+b(\deg) (d-1)} - 2^{d+b(\deg) (d-1)} \\
&= 2 \cdot 2^{d+b(\deg) (d-1)} - 2^{d+b(\deg) (d-1)}.
\end{align*}
\]

A.24 Proof of Lemma 7.8

Proof. In the following, let \(\vec{u} = A \cdot \vec{x}\).

\(\Leftarrow\Rightarrow\). Let \(\succ\) be a total ordering and \(\lambda\) the unique eigenvalue of \(A\), i.e., the diagonal of \(A\) only contains \(\lambda\). We now prove that the matrix \(A - \lambda \cdot I^{d \times d}\) has rank at least \(d - 1\), i.e., its kernel, which is the eigenspace of \(A\) w.r.t. \(\lambda\), has dimension at most 1. As \(\lambda\) is an eigenvalue of \(A\), its eigenspace then must be exactly one-dimensional.

Since \(A\) is triangular, so is \(A - \lambda \cdot I^{d \times d}\). As \(\succ\) is a total ordering, the super-diagonal of \(A - \lambda \cdot I^{d \times d}\) contains only non-zero values, whereas its diagonal contains only zeros. Thus, by deleting the first row and the last column of this matrix, we obtain a triangular \((d - 1) \times (d - 1)\) submatrix \(B\) whose diagonal is the super-diagonal of \(A - \lambda \cdot I^{d \times d}\), i.e., it contains only non-zero values. Thus, the product of the diagonal entries of \(B\) is non-zero, i.e., \(\det(B) \neq 0\). But \(B\) is a submatrix of \(A - \lambda \cdot I^{d \times d}\), i.e., \(A - \lambda \cdot I^{d \times d}\) has a non-zero \((d - 1) \times (d - 1)\) minor. Hence, \(\text{rank}(A - \lambda \cdot I^{d \times d}) \geq d - 1\).

\(\Rightarrow\). Let us assume that \(\succ\) is not a total ordering. In this case, the super-diagonal of \(A - \lambda \cdot I^{d \times d}\) contains a zero and its diagonal contains only zeros. Due to triangularity, its rank can be at most \(d - 2\). Thus, \(A\) has at least two linear independent eigenvectors. \(\square\)
A.25 Proof of Lemma 7.12

Proof. Let $Q$ be the Jordan normal form of $A$. From the form of $Q^n_\lambda$ in Fig. 9, we directly obtain the following observation:

For all $\lambda \in \text{spec}(A)$ and $1 \leq r \leq \nu(\lambda)$, the $(\text{idx}(\lambda) + r)$-th element of $\vec{q} = Q^n \cdot \vec{x}$ is

$$\sum_{s=r}^{n} \left\{ \binom{n}{s-r} \cdot \lambda^{n-s+r} \cdot x_{\text{idx}(\lambda)+s} \right\}.$$  \hfill (22)

For $s - r \in \mathbb{N}$, in general $\binom{n}{s-r}$ is not a polynomial in the variable $n$, as one has to distinguish the cases $n < s - r$ and $n \geq s - r$. But since our approach from Sect. 4 analyzes eventual non-termination, we are only interested in validity of formulas for large enough $n$. Thus, we may assume $n \geq s - r$. Then, $\binom{n}{s-r}$ is indeed a polynomial from $\mathbb{Q}[n]$ of degree $s - r$, i.e., there are coefficients $c_{s-r,j} \in \mathbb{Q}$ such that for all $n \geq s - r$ we have $\binom{n}{s-r} = \sum_{j=0}^{s-r} c_{s-r,j} \cdot n^j$.

In fact, $c_{s-r,j} = \frac{\text{stir}(s-r,j)}{(s-r)!}$ where stir is the signed Stirling number of the first kind (see, e.g., [27, Ch. 6]). While stir’s formal definition is not of interest for us, we use $\text{stir}(s-r,s-r)$ to be the Jordan normal form of $\vec{q}$, where the coefficient $c_{s-r,j}$ is a linear polynomial in the variables $x_{\text{idx}(\lambda)+s}, \ldots, x_{\text{idx}(\lambda)+\nu(\lambda)}$. Note that the coefficient of $x_{\text{idx}(\lambda)+s}$ in $\alpha_s$ is $c_{s-r,s-r} = \frac{\text{stir}(s-r,s-r)}{(s-r)!} \cdot \lambda^{s-r} = \frac{\text{stir}(s-r,s-r)}{(s-r)} \cdot \lambda^{s-r} \neq 0$.

By (23) and (24), the elements of $\vec{q}$ are hierarchical expressions. Note that here we indeed need $\text{stir}(s-r,s-r) = 1 \neq 0$, i.e., $\deg x_{\text{idx}(\lambda)+s}(\alpha_s) = 1$.

A.26 Proof of Lemma 7.19

Proof. We have

$$p = f(0, \ldots, 0) + \sum_{1 \leq i \leq k, r \in F(i)} \text{coeff}(f, x_r) \cdot h_r.$$  \hfill (25)

Moreover, for every $1 \leq i \leq k$ and every $r \in B_i$, the hierarchical expression $h_r$ has the form

$$h_r = \sum_{s=0}^{n_1+\ldots+n_i} \beta_{\alpha_s} \cdot \lambda^\alpha_s,$$

where $\beta_{\alpha_s} \in \mathbb{Q}[x_1, \ldots, x_{n_1+\ldots+n_i}]$. \hfill (26)

(d) Here, $\lambda_i \neq b$ for all $1 \leq i \leq k$ or $\lambda_i = b$ for some $1 \leq i \leq k$ but $F(i) = \emptyset$. Moreover, $b \neq 1$ or $a \neq 1$. Hence, (25) and (26) imply that there is no $\alpha_s$ with $b_s = b$ and $a_s = a$.

(a) If $\lambda_i \neq 1$ for all $1 \leq i \leq k$ or $\lambda_i = 1$ for some $1 \leq i \leq k$ and $F(i) = \emptyset$, then in (d) we already showed that there is no $\alpha_s$ with $b_s = 1$ and $a_s = 1$. However, there is an $\alpha_s$ with $b_s = 1$ and $a_s = 0$, viz. $\alpha_s = f(0, \ldots, 0)$. Thus, we have $\alpha_s = 0$ if $f(0, \ldots, 0) = 0$.

(c) The largest number in $B_i$ is $n_1 + \ldots + n_i$. So if $a > n_1 + \ldots + n_i - \min F(i)$ then we have $\alpha_s = 0$. This is sound because then there is no $\alpha_s$ with $b_s = a$ and $a_s = a$.

Hence, we now consider the case $a \leq n_1 + \ldots + n_i - \min F(i)$. Since $\lambda_i = b$ and $F(i) \neq \emptyset$, there is an $\alpha_m$ with $b_m = b$. Let $1 \leq s_0 \leq \ell$ be the largest number with $b_{s_0} = b = \lambda_i$. By (25) and (26), the corresponding addend $\alpha_{s_0} \cdot n^{s_0} \cdot b_{s_0}^m$ of $p$ has the form $\text{coeff}(f, x_{s_0}) \cdot \beta_{\alpha_{s_0}} \cdot \lambda_{s_0}^\alpha_{s_0}$ for the smallest possible $r \in F(i)$. So we get

$$\alpha_{s_0} \cdot n^{s_0} \cdot b_{s_0}^m = \text{coeff}(f, x_{\min F(i)}) \cdot \beta_{\min F(i), n_1+\ldots+n_i} \cdot n^{s_0} \cdot s_0 - \min F(i) \cdot \lambda_i^s.$$
Here, $\beta_{\min} F(i), x_1 + \ldots + x_i \in Q S[x_1 + \ldots + x_i]_{\text{lin}}$ and in fact, $\beta_{\min} F(i), x_1 + \ldots + x_i = c x_1 + \ldots + x_i$ for some $c \neq 0$. Hence, $\alpha_{s_0} = 0$ is equivalent to $x_1 + \ldots + x_i = 0$.

Now consider the second largest number $s_1 = s_0 - 1 \leq \ell$ such that $b_{s_1} = b = \lambda_i$. By (25) and (26), for $a_{s_1} = 1 + \ldots + v_i - \min F(i) - 1$ we obtain

$$\alpha_{s_1} \cdot n^{s_1} \cdot b_{s_1} = \sum_{r \in F(i), \min F(i) \leq r < \min F(i) + 1} \text{coeff}(f, x_r) \cdot \beta_{r, r + a_{s_1}} \cdot n^{a_{s_1}} \cdot \lambda^n,$$

where $\beta_{r, r + a_{s_1}} \in Q S[x_{r + a_{s_1}} + \ldots + x_1 + \ldots + x_i]_{\text{lin}}$. By taking into account that $x_1 + \ldots + x_i = 0$, this simplifies to

$$\alpha_{s_1} = \beta_{r, r + a_{s_1}} \cdot \lambda^n,$$

where $\beta_{r, r + a_{s_1}} \in Q S[x_{r + a_{s_1}} + \ldots + x_1 + \ldots + x_i]_{\text{lin}}$. By taking into account that $x_1 + \ldots + x_i = 0$, this simplifies to

$$\alpha_{s_1} = \beta_{r, r + a_{s_1}} \cdot \lambda^n.$$

Here, $\beta_{\min} F(i), x_1 + \ldots + v_i - 1 \in Q S[x_1 + \ldots + v_i - 1, x_1 + \ldots + x_i]_{\text{lin}}$. But when again taking into account that $x_1 + \ldots + v_i = 0$, in fact we have $\beta_{\min} F(i), v_i + v_i - 1 = \epsilon' x_1 + \ldots + v_i - 1$ for some $\epsilon' \neq 0$. Hence, $\alpha_{s_0} = 0$ is equivalent to $x_1 + \ldots + v_i = 0$, or in other words (as $\nu_1 + \ldots + v_i - 1 = \min F(i) + a_{s_2} = 0$),

We repeat this reasoning until we reach an $s' \leq \ell$ with $a_{s'} = a$. Thus, $\alpha_{s_0} = 0$ for all $s \leq \ell$ with $b_s = b = \lambda_i$ and $a_s \geq a$ is equivalent to $x_j = 0$ for all $min F(i) + a \leq j \leq v_i + v_i$, i.e., to $\bigwedge r \in B_i, a + \min F(i) \leq j \ (x_j = 0)$ according to (c).

If $b_s = 1$ and $a_s = 0$, then (25) and (26) imply

$$\alpha_s = f(0, \ldots, 0) + \sum_{r \in F(i)} \text{coeff}(f, x_r) \cdot \beta_{r, r},$$

where $\beta_{r, r} \in Q S[x_1, x_1 + \ldots + x_i]_{\text{lin}}$. Clearly $r \in F(i)$ implies $r \geq \min F(i)$. Taking into account that $x_j = 0$ for all $min F(i) \leq j \leq v_i + v_i$, we therefore obtain

$$\alpha_s = f(0, \ldots, 0) + \text{coeff}(f, x_{\min F(i)}) \cdot \beta_{\min F(i), \min F(i)},$$

where $\beta_{\min F(i), \min F(i)} = \epsilon_{\min F(i)} \cdot x_{\min F(i)}$. Therefore,

$$\alpha_s = 0 \iff f(0, \ldots, 0) + \text{coeff}(f, x_{\min F(i)}) \cdot \epsilon_{\min F(i)} \cdot x_{\min F(i)} = 0 \iff x_{\min F(i)} = M.$$

A.27 Proof of Lemma 7.21

Proof. (a) As in the proof of Case (a) of Lemma 7.19, we have $\alpha_{s_0} = f(0, \ldots, 0)$ and thus $\alpha_{s_0} > 0 \iff f(0, \ldots, 0) > 0$. In this case, for all $s_0 \leq \ell$ we must have $b_s > 1$. Hence, $\bigwedge_{s_0 \leq s \leq \ell} \alpha_s = 0 \iff \bigwedge_{s_0 \leq s \leq \ell} \lambda_s = 0$ by Lemma 7.19.

(b) Now $(b_{s_0}, a_{s_0}) = (1, 0)$, $\lambda_{s_0} = 1$ for some $1 \leq i_0 \leq k$, and $F(i_0) \neq \emptyset$. By Lemma 7.19 we obtain that $\alpha_{s_0} = 0$ for all $s_0 < s \leq \ell$ with $b_s = 1$ is equivalent to $\lambda_{s_0 + 1} = 0$. Finally, as in the proof of Case (b) of Lemma 7.19, we get

$$\alpha_{s_0} = f(0, \ldots, 0) + \text{coeff}(f, x_{\min F(i_0)}) \cdot \beta_{\min F(i_0), \min F(i_0)},$$

with $\beta_{\min F(i_0), \min F(i_0)} = \epsilon_{\min F(i_0)} \cdot x_{\min F(i_0)}$. For $C = \text{coeff}(f, x_{\min F(i_0)}) \cdot \epsilon_{\min F(i_0)}$,

$$\alpha_{s_0} > 0 \iff f(0, \ldots, 0) + C \cdot x_{\min F(i_0)} > 0 \iff C \cdot x_{\min F(i_0)} + \sum_{\{0, 1, \ldots, 0\}} > 0 \iff \text{sign}(C) \cdot x_{\min F(i_0)} + \sum_{\{0, 1, \ldots, 0\}} > 0.$$
Lemma 7.19

Alg. 4

Thm. 7.36

Proof. Soundness: We show that \( \alpha_s = 0 \) for all \( s \geq 1 \) and \( F(i_0) = \emptyset \), then \( \mathcal{N}^k_{s=0} (\alpha_s = 0) \) is false. The reason is that \( \text{coefs}(p) \) contains \( a_s^{(1,0)} \) with \( s_0 < s \) and since there is either no \( \alpha_s = 1 \), or \( \lambda_{i_0} = 1 \) for some \( 1 \leq i_0 \leq k \) but \( F(i_0) = \emptyset \), we have \( \alpha_s = 0 \).

(d) In the remaining case, since \( b_{s_0} \neq 1 \), there must be an \( 1 \leq i_0 \leq k \) with \( \lambda_{i_0} = b_{s_0} \) and \( F(i_0) \neq \emptyset \). By Lemma 7.19, \( \alpha_s = 0 \) for all \( s_0 < s \leq \ell \) with \( b_s = b_{s_0} = \lambda_{i_0} \) is equivalent to \( \text{zero}(\lambda_{i_0}, a_{s_0} + 1) \), and \( \alpha_s = 0 \) for all \( s_0 < s \leq \ell \) with \( b_s \neq b_{s_0} \) is equivalent to \( \mathcal{N}^k_{s=0} \text{zero}(\lambda_s,0) \). Finally, (25) and (26) from the proof of Lemma 7.19 imply

\[
\alpha_{s_0} = \sum_{r \in F(i_0), r \leq v_1 + \ldots + v_{i_0} - a_{s_0}} \text{coeff}(f, x_r) \cdot \beta_{r,r+a_{s_0}} \cdot n^{s_0} \cdot \lambda_{i_0}^{s_0},
\]

for \( \beta_{r,r+a_{s_0}} \in \mathbb{Q}[x_{r+a_{s_0}}, \ldots, x_{v_1 + \ldots + v_{i_0} + 1}] \). Clearly \( r \in F(i_0) \) implies \( r \geq \min F(i_0) \). Considering that \( x_j = 0 \) for all \( \min F(i_0) + a_{s_0} < j \leq v_1 + \ldots + v_{i_0} \), we therefore obtain

\[
\alpha_{s_0} = \text{coeff}(f, x_{\min F(i_0)}) \cdot \beta_{\min F(i_0), \min F(i_0) + a_{s_0}} \cdot n^{s_0} \cdot \lambda_{i_0}^{s_0},
\]

where \( \beta_{\min F(i_0), \min F(i_0) + a_{s_0}} = \text{min}_F(i_s) + a_{s_0} \cdot \text{min}_F(i_0) + a_{s_0} \). Let \( sg = \text{sgn}(\text{coeff}(f, x_{\min F(i_0)})) \cdot \text{min}_F(i_0) + a_{s_0} \).

Thus, \( \alpha_{s_0} > 0 \) is equivalent to \( sg \cdot \text{min}_F(i_0) + a_{s_0} > 0 \).

A.28 Proof of Lemma 7.32

Proof. For any \( 0 \leq s \leq \ell \) we have \( \text{cnAssg}(p_{f,s}) \subseteq cA \), where

\[
cA = \{ v \in \{0, 1, \ldots, \star\}^d \mid \forall 1 \leq i \leq k, \text{there is at most one } j \in B_i \text{ with } v_j \neq 0 \}.
\]

Now we over-approximate the cardinality of \( cA \). If \( v \in cA \), then for every \( 1 \leq i \leq k \), we have \( v_j = 0 \) for at most one \( j \in B_i \). So for the values \( v_j \) with \( j \in B_i \), there are \( 3 \cdot |B_i| + 1 \) possibilities: exactly one of them is \( 1, -1, \) or \( \star \), or we have \( v_j = 0 \) for all \( j \in B_i \). When combining this result for all \( 1 \leq i \leq k \) by multiplication, we obtain \( |\text{cnAssg}(p_{f,m})| \leq (3 \cdot |B_i| + 1)^k \).

A.29 Proof of Cor. 7.34

Proof. We show that \( \ell \leq d + 1 \) for \( |\text{coefs}(p)| = \ell \). Then the result follows from Lemma 7.32 and the definition of \( \text{cnAssg} \), since \( \text{cnAssg}(\text{ic}(p \circ 0)) \subseteq \bigcup_{i=0}^{\ell} \text{cnAssg}(p_{f,i}) \). The number of coefficients of \( p \) is determined by the number of terms of the form \( \alpha \cdot n^a \cdot b^n \) occurring in \( p \). But these terms are connected to the bases \( 0 < \lambda_1 < \ldots < \lambda_k \) of the hierarchical partition: either \( b = \lambda_i \) for some \( 1 \leq i \leq k \) and \( 0 < a \leq v_i - 1 \) or \( a = 0 \) and \( b = 1 \). So we have at most \( 1 + \sum_{i=1}^{k} v_i = 1 + d \) such terms in \( p_i \), i.e., \( \ell \leq d + 1 \).

A.30 Proof of Thm. 7.36

Proof. Soundness: If Alg. 4 returns \( \top \), then there is a \( \vec{v} \in \text{cnAssg}(\text{ic}(\varphi(\vec{h}))) \) where \( \text{ic}(\varphi(\vec{h}))) \) is satisfiable, i.e., a \( w \in S \) where \( \vec{v}[\star / w] \) is a model of \( \text{ic}(\varphi(\vec{h})) \). So \( \exists \vec{x} \in S^d, \text{ic}(\varphi(\vec{h})) \) is valid.

Completeness: Let \( \psi = \text{ic}(\varphi(\vec{h})) \) and let \( \exists \vec{x} \in S^d, \psi \) be valid, i.e., there is a \( \vec{v} \in S^d \) such that \( \psi(\vec{v}) \) holds. We have to prove that there is a \( \vec{v} \in \text{cnAssg}(\text{ic}(\varphi(\vec{h}))) \) where \( \psi(\vec{v}) \) is satisfiable. Then the claim follows as the algorithm calls SMT on \( \psi(\vec{v}) \) and thus returns \( \top \).
Note that \( \phi \) is a propositional formula only built from the connectives \( \land \) and \( \lor \), i.e., it does not contain \( \neg \). By construction, \( \text{ic}(\phi(h)) \) results from \( \phi \) by replacing each atom \( f \rightarrow 0 \) by \( \text{ic}(f(h)) \lor 0 \). So similar to the concept of fundamental sets in the proof of Thm. 4.9, there is a subset \( \{ \text{ic}(f_1(h)) \lor v_1 \), \ldots, \( \text{ic}(f_r(h)) \lor v_r \} \) of these formulas such that \( \overline{v} \) satisfies them all and such that satisfying these formulas is sufficient for satisfying \( \phi \).

Let \( t_1, \ldots, t_\ell \) be the numbers of coefficients in the poly-exponential expressions \( f_1(h) \), \( \ldots, f_r(h). \) By Cor. 7.25, \( \text{ic}(f_j(h)) \lor v_r \) has the form \( \bigvee_{s=1}^{t_j} \rho_{f_j,s} \) or \( \bigvee_{s=1}^{t_j} \rho_{f_j,s} \) for each \( 1 \leq r \leq e \). So for every \( r \) there is at least one \( s \) where \( \rho_{f_j,s}(\overline{v}) \) is true. But due to the construction of \( \rho_{f_j,s} \) in Lemma 7.21 and Cor. 7.23, for every \( \overline{v} \) there is at most one \( 0 \leq s \leq t_j \) where \( \rho_{f_j,s}(\overline{v}) \) is true. Thus, if for every \( 1 \leq r \leq e \), there is a unique \( 0 \leq s_r \leq t_r \) where \( \rho_{f_r,s_r}(\overline{v}) \) is true.

Let \( B_1, \ldots, B_k \) again be the blocks from the \( k \)-partition \( h_1, \ldots, h_k \). Note that if there is a block \( B_i \) and some \( \rho_{f_j,s} \), requires \( v_j \) with \( j \in B_i \) to be non-zero, then \( \rho_{f_j,s} \) requires all \( v_j \) with \( j' > j \) and \( j' \in B_i \) to be zero. Thus, since \( \overline{v} \) satisfies all formulas \( \rho_{f_r,s_r} \), for \( 1 \leq r \leq e \), each \( B_i \) there is at most one \( r \in B_i \) where some \( \rho_{f_r,s_r} \) requires \( v_j \) to be non-zero. Hence, we can assume that for each block \( B_i \) there is at most one \( j \in B_i \) where \( v_j \neq 0 \).

(i) Case \( \lambda_{i0} = 1 \) for some \( 1 \leq i_0 \leq k \): In this case, for the non-zero entries in \( \overline{v} \) belonging to indices in \( B_i \) with \( i \neq i_0 \), only their sign is important since the formulas requiring them to be non-zero are interval conditions according to Def. 7.27 (b) resulting from Lemma 7.21 (d). So these entries can be chosen to be 1 or -1.

If there is a (unique) non-zero value \( v_j \) with \( j \in B_{i_0} \), its value is indeed important: the formulas requiring this value to be non-zero have the form of Def. 7.27 (e) (in Lemma 7.21 (b)) or Def. 7.27 (d) (in Lemma 7.21 (d)) or Def. 7.27 (c) (in Cor. 7.24).

Thus, let

\[
v_j' = \begin{cases} 
0, & \text{if } v_j = 0 \\
*, & \text{if } v_j \neq 0 \land j \in B_{i_0} \\
1, & \text{if } v_j > 0 \land j \notin B_{i_0} \\
-1, & \text{if } v_j < 0 \land j \notin B_{i_0} 
\end{cases}
\]

As discussed before, \( \psi(\overline{v}') \) is satisfiable as \( (\rho_{f_1,s_1} \land \ldots \land \rho_{f_r,s_r})(\overline{v}') \) is satisfiable by construction: If the formula contains *, then assigning * the unique non-zero value \( v_j \neq 0 \) for \( j \in B_{i_0} \) is a satisfying assignment.

(ii) Case \( \lambda_{i0} \neq 1 \) for all \( 1 \leq i_0 \leq k \): In this case, \( \rho_{f_1,s_1}, \ldots, \rho_{f_r,s_r} \) are interval conditions according to Def. 7.27 (a) or (b). Thus, only the sign of the non-zero values in \( \overline{v} \) is important to satisfy these formulas. Hence we define

\[
v_j' = \begin{cases} 
0, & \text{if } v_j = 0 \\
1, & \text{if } v_j > 0 \\
-1, & \text{if } v_j < 0 
\end{cases}
\]

Thus, \( \psi(\overline{v}') \) is satisfiable since \( (\rho_{f_1,s_1} \land \ldots \land \rho_{f_r,s_r})(\overline{v}') \) is true by construction.

So in both cases \( \overline{v}' \in \bigcup_{1 \leq r \leq e} \text{cndAssg}(\rho_{f_r,s_r}) \subseteq \text{cndAssg}(\psi) \), i.e., Alg. 4 returns \( \top. \)