The L-depth Eventual Linear Ranking Functions for Single-path Linear Constraint Loops

Yi Li, Guang Zhu and Yong Feng
Key Laboratory of Automated Reasoning and Cognition
CIGIT, CAS, Chongqing, China 400714
Email: liyi@cigit.ac.cn, zhuguang@cigit.ac.cn, yongfeng@cigit.ac.cn

***Abstract*—Termination of loop programs has received extensive attention in these years. In this paper, we focus on the termination of single-path linear constraint loops. For single-path linear constraint loops which have no linear ranking functions or eventual linear ranking functions, we present a complete method to detect the existence of $l-$depth eventual linear ranking functions. Our method extends the work of Bagnara and Mesnard. The prototype of our method has been implemented and the effectiveness of our method has been shown by experimental results.**

Keywords– Termination Analysis, L-depth Eventual Linear Ranking Functions, Single-path Linear Constraint Loop Programs, RegularChains

I. Introduction

Termination analysis of loop programs is endowed with a great importance for software correctness. One popular method for termination analysis is testing the existence of ranking functions. For rational or integer linear loops, synthesis of ranking functions has been widely studied in [3], [6], [7], [8], [9], [10], [11], [13], [14], [16], [18]. The complexity of the linear ranking function problem for linear loops is analysed in[4], [5], [6], [7].

In 2001, Colón and Sipma [15] synthesized linear ranking functions (LRFs for short) to prove loop termination by the technique of polyhedral. For single-path linear loops, Podelski and Rybalcheko [22] first proposed a complete and efficient method for synthesizing LRFs based on linear programming when program variables range over the reals and rationals in 2004. Their method is dependent on Farkas' lemma which provides a technique to extract hidden constraints from a system of linear inequalities.

Bradley et al.[8], [9] extended the work presented in [15] and showed how to synthesize lexicographic LRFs with linear supporting invariants over multi-path linear constraint loops in 2005. In [14], Chen et al. gave a technique to generate non-linear ranking functions for polynomial programs by solving semi-algebraic systems. Cook et al. [16] described an automatic method for finding sound under-approximations to weakest preconditions to termination.

In 2012, Chen et al. [13] characterized a method to generate proofs of universal termination for linear simple loops based on the synthesis of disjunctive ranking relations. Their method is a generalization of the method given in [22]. In [18], a method was proposed by Ganty and Genaim to partition the transition relations, which can be applied to conditional termination analysis. Bagnara et al. [3] analysed termination of single-path linear constraint loops by the existence of eventual LRFs, where the eventual LRFs are linear functions that become ranking functions after a finite unrolling of the loop. In 2013, Cook et al. [17] presented a method for proving termination by Ramsey-based termination arguments instead of lexicographic termination arguments. For lasso programs, Heizmann et al. suggested a series of techniques to synthesize termination arguments in [19], [20].

In this paper, we focus on the termination of single-path linear constraint (SLC for short) loops. By means of the definition of eventual ranking functions given by Bagnara and Mesnard in [3] and the technique due to Podelski and Rybalchenkopresent in [22], we present a new method for synthesizing $l-$depth eventual LRFs. For synthesizing an eventual LRFs, one need to construct a increasing function first, and then divide the region specified by the loop guards into two subregions, a LRF over one of subregions is such eventual LRF. In our method, we generalize the definition of eventual LRF to l-depth eventual LRFs. Obviously, l-depth eventual LRFs are eventual LRFs if l=1. It will be shown that for a given SLC loop, the loop is terminating if such an l-depth eventual LRF exists.

Experimental results show that for some SLC loops, whose LRFs or eventual LRFs fail to be found by the methods in [3], [7], [8], [15], our method can find the desired $l-$depth eventual LRFs for such SLC loops to be terminating. In this paper, we use the tool RegularChains [12] in Maple to eliminate quantifiers.

The rest of the paper is organized as follows. Section 2 introduces necessary definitions and background information regarding LRFs. In Section 3, a technique based on Podelski and Rybalchenkopresent is introduced to synthesize linear increasing functions(LIFs for short) and eventual LRFs. In Section 4, an algorithm is presented to find $l-$depth eventual LRFs for SLC loops, and we give an example to illustrate our methods. Section 5 concludes the paper.

II. Preliminaries

In this paper, we focus on the termination of *single-path linear constraint*(SLC) loops, where the loop guard is a conjunction of linear inequalities and the loop body is the update of the variables in an affine linear way.

A linear function $\rho : \mathbb{Q}^n \Rightarrow \mathbb{Q}$ is of the form $\rho(\mathbf{x}) = \mathbf{r}^T \cdot \mathbf{x}$, where $\mathbf{r} \in \mathbb{Q}^n$. Ranking functions ensure termination if the function maps a vector to an element of some well-founded set such that the execution of the body of the loop causes the value of the function to decrease in order.

First, we recall the definition of single-path linear constraint loop P given in [3]

$$\textit{while}(\mathbf{B}\mathbf{x} \leq \mathbf{b}) \quad \textit{do} \quad \mathbf{x}' \leq \mathbf{A}\mathbf{x} + \mathbf{v}, \tag{1}$$

where $\mathbf{x} = (x_1, ..., x_n)^T$ and $\mathbf{x}' = (x'_1, ..., x'_n)^T$ are column vectors of variables. Suppose that there exists p inequations in loop guard and $p > 0$, then $\mathbf{B} \in \mathbb{Q}^{p\times n}, \mathbf{A} \in \mathbb{Q}^{n\times n}, \mathbf{b} \in \mathbb{Q}^p, \mathbf{v} \in \mathbb{Q}^n$. The constraint $\mathbf{B}\mathbf{x} \leq \mathbf{b}$ is the loop guard and $\mathbf{x}' \leq A\mathbf{x}+\mathbf{v}$ is called the loop body. The loop body is the form of a simultaneous nondeterministic update. Define $\tau(\mathbf{x}, \mathbf{x}') \stackrel{def}{=} \{\mathbf{B}\mathbf{x} \leq \mathbf{b} \wedge \mathbf{x}' \leq \mathbf{A}\mathbf{x} + \mathbf{v}\}$. Let $\Omega \stackrel{def}{=} \{\mathbf{x} \in \mathbb{Q}^n | \mathbf{B}\mathbf{x} \leq \mathbf{b}\}$, $\overline{\Omega} = \mathbb{Q}^n \backslash \Omega$.

For this kind of loop program such as Formula (1), it can be represented by the following system:

$$(\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}. \tag{2}$$

For loop program (1), set $\mathbf{x}' \leq \mathbf{A}\mathbf{x} + \mathbf{v}$. Then in System (2), $\mathbf{H} = \begin{pmatrix} \mathbf{B} \\ -\mathbf{A} \end{pmatrix} \in \mathbb{Q}^{(p+n)\times n}$, $\mathbf{H}' = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \in \mathbb{Q}^{(p+n)\times n}$, $\mathbf{c} = \begin{pmatrix} \mathbf{b} \\ \mathbf{v} \end{pmatrix} \in \mathbb{Q}^{(p+n)}$, where $\mathbf{I}$ is n identity matrix.

Definition 1. ***(Linear ranking functions for Single-path Linear Constraint Loops)*** *Given a single-path linear constraint loop P, let $\rho(\mathbf{x})$=$\mathbf{r}^T \cdot \mathbf{x}+c$ be a linear function where $\mathbf{r} \in \mathbb{Q}^n$ and $c \in \mathbb{Q}$. Then $\rho(\mathbf{x})$ is a linear ranking function over Ω if the following formula holds.*

$$\forall \mathbf{x}, \mathbf{x}', \ \big(\tau(\mathbf{x}, \mathbf{x}') \Rightarrow \rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1\big). \tag{3}$$

It is easy to see that the existence of linear ranking functions for P implies that program P is terminating.

Note that the decrease by 1 in the formula of Definition 1 can be replaced by any positive number δ. For example, for a given loop, assume $\rho(\mathbf{x})$ satisfies the following formula:

$$\forall \mathbf{x}, \mathbf{x}', \ \big(\tau(\mathbf{x}, \mathbf{x}') \Rightarrow \rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq \delta > 0\big).$$

Then, it is easy to see that $\frac{1}{\delta}\rho(\mathbf{x})$ is a linear ranking function for the loop according to Definition 1.

Next, we introduce the following Theorem 1 given by Podelski et al.

Theorem 1. ***(Synthesis of Linear Ranking Functions)*** *Given a single-path linear constraint loop program $(\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$, a linear ranking function exists iff there exist non-negative vectors over the rationals λ_1 and λ_2 such that the following system is satisfiable.*

$$S_\rho : \begin{cases} \lambda_1, \lambda_2 \geq 0, \\ \lambda_1 \mathbf{H}' = 0, \\ (\lambda_1 - \lambda_2)\mathbf{H} = 0, \\ \lambda_2(\mathbf{H} + \mathbf{H}') = 0, \\ \lambda_2 \mathbf{c} < 0. \end{cases} \tag{4}$$

Given λ_1 and λ_2, solutions of System (4), by Theorem 1, one can construct a linear ranking function $\rho(\mathbf{x}) = \mathbf{r}^T \cdot \mathbf{x}$, where $\mathbf{r} = \lambda_2 \mathbf{H}'$.

However, for some loop programs, System (4) has no solutions, that is LRFs do not exist. Therefore, for such loops, their termination remains unknown.

III. Eventual Linear Ranking Functions for SLC Loops

In this part, we first introduce the definitions of linear increasing functions(LIFs for short) and eventual linear ranking functions, which is first given by Bagnara et al. in [3]. And then we establish Theorem 3, which tell us that a SLC loop has eventual LRFs iff a certain semi-algebraic system has solutions over the rationals. Theorem 2 and 3 are different from the methods in [3] for finding LIFs and eventual LRFs, since the latter finds LIFs and eventual LRFs by Farkas' Lemma.

Definition 2. ***(Linear Increasing Functions for SLC loops[3])*** *Given a single-path linear constraint loop P, let $f(\mathbf{x}) = \mathbf{d}^\mathbf{T} \cdot \mathbf{x}$ be a linear function where $\mathbf{d} \in \mathbb{Q}^n$. We call $f(\mathbf{x})$ a linear increasing function over Ω if the following formula holds.*

$$\forall \mathbf{x}, \mathbf{x}', (\tau(\mathbf{x}, \mathbf{x}') \Rightarrow f(\mathbf{x}') - f(\mathbf{x}) \geq 1). \tag{5}$$

Let $\Psi = \{\mathbf{d} \in \mathbb{Q}^n | \forall \mathbf{x}, \mathbf{x}', (\tau(\mathbf{x}, \mathbf{x}') \Rightarrow f(\mathbf{x}') - f(\mathbf{x}) \geq 1)\}$ be the space of LIFs. For some $k \in \mathbb{Q}$, set $\Omega_f = \{\mathbf{x} \in \mathbb{Q}^n | f(\mathbf{x}) \geq k\}$.

Likewise, the number 1 in Formula (5) can be replaced by a positive number δ.

Remark 1. *Given a single-path linear constraint loop P, suppose that $f(\mathbf{x})$ is a linear increasing function over Ω and $\Omega \cap \Omega_f \neq \varnothing$. Since $f(\mathbf{x})$ is increasing over Ω, for any $\mathbf{x} \in \Omega - \Omega_f$, $\mathbf{x}$ must fall into $\overline{\Omega - \Omega_f} = \overline{\Omega} \cup (\Omega \cap \Omega_f)$ after a finite number of iterations, that is there does not exist infinite iterative sequences over $\Omega - \Omega_f$. Because if there exists a infinite iterative sequence $S = \{\mathbf{x_0}, \mathbf{x_1}, ...\}$ over $\Omega - \Omega_f$, then we have $f(\mathbf{x_i}) < k$ since $S \subseteq \Omega - \Omega_f \subseteq \overline{\Omega_f}$. And since $f(\mathbf{x})$ is strictly increasing over Ω, we have $f(\mathbf{x_i}) - f(\mathbf{x_{i-1}}) \geq 1$ for all $i = 1, 2, ...$ Therefore, there must exist a positive integer N, such that $f(\mathbf{x_N}) \geq k$. This contradicts the above assumption. Hence, there does not exist infinite iterative sequences over $\Omega - \Omega_f$. Likewise, for any $\mathbf{x} \in \Omega \cap \Omega_f$, because $f(\mathbf{x})$ is increasing over Ω, after an arbitrary number of iterations, $\mathbf{x}$ cannot fall into $\overline{\Omega_f}$. Furthermore, for any $\mathbf{x} \in \Omega \cap \Omega_f$, $\mathbf{x}$ cannot fall into $\Omega - \Omega_f$, after an arbitrary number of iterations, since $\Omega - \Omega_f \subseteq \overline{\Omega_f}$.*

By the definition of LIF, we can synthesize eventual linear ranking functions for single-path linear constraint loops, which is presented first in [3].

Definition 3. ***(Eventual Linear Ranking Functions for SLC Loops[3])*** *Given a single-path linear constraint loop P, let $f(\mathbf{x})$ be a linear increasing function over Ω. A linear function $\rho(\mathbf{x})$ is an eventual linear ranking function over Ω if the following formula holds.*

$$\begin{aligned}\exists k, \forall \mathbf{x}, \mathbf{x}', \ \big(\tau(\mathbf{x}, \mathbf{x}') \wedge f(\mathbf{x}) \geq k) \Rightarrow \\ \rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1\big)\end{aligned} \tag{6}$$

According to the arguments in [3], we know that if there is an eventual LRF for the loop program P, then P is terminating.

Remark 2. *Given a single-path linear constraint loop P, if $\rho(\mathbf{x})$ is an eventual linear ranking function over Ω, then $\rho(\mathbf{x})$ is a linear ranking function over $\Omega \cap \Omega_f$.*

Example 1. *Consider the loop program P_1 as follows:*

$$\textit{while}(x_1 \geq 0) \quad \{\ x_1' \leq x_1 + x_2; x_2' \leq x_2 - 1;\ \}$$

We fail to synthesize linear ranking functions by Theorem 1. Next, we test the existence of eventual linear ranking functions.

By Definition 2, a LIF $f(\mathbf{x}) = -x_2$ can be gotten for loop program P_2. Then we can synthesize a satisfiable eventual linear ranking function $\rho(\mathbf{x}) = x_1$ by Definition 3. Hence the loop program P_1 always terminates. By the above definitions, $\Omega = \{\mathbf{x} \in \mathbb{Q}^n | x_1 \geq 0\}$ and $\Omega_f = \{\mathbf{x} \in \mathbb{Q}^n | -x_2 \geq k\}$. For $f(\mathbf{x}) \geq k$, we get $f(\mathbf{x}) = -x_2$ and $k > 0$ by computing. Let $k = 1$, then $\Omega \cap \Omega_f = \{\mathbf{x} \in \mathbb{Q}^n | x_1 \geq 0 \wedge x_2 < -1\}$.

We know that $\rho(\mathbf{x}) = x_1$ is an eventual linear ranking function over Ω by computing. After testing, $\rho(\mathbf{x}) = x_1$ is a linear ranking function over $\Omega \cap \Omega_f$.

Similar to Theorem 1, we establish a necessary and sufficient condition, which is a semi-algebraic system to test the existence of LIFs.

Theorem 2. ***(Test Existence of Linear Increasing Functions)*** *Given a single-path linear constraint loop program $(\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$, a linear increasing function exists iff there exists a non-negative vector μ over the rationals such that the following system is satisfiable.*

$$\mu \geq 0 \ \wedge \ \mu(\mathbf{H} + \mathbf{H}') = 0 \ \wedge \ \mu\mathbf{c} < 0. \tag{7}$$

Proof. Given an SLC loop program $P : (\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$ where $\mathbf{H} \in \mathbb{Q}^{m \times n}$, $\mathbf{H}' \in \mathbb{Q}^{m \times n}$, $\mathbf{x} \in \mathbb{Q}^n$ and $\mathbf{x}' \in \mathbb{Q}^n$. Denote by $\mu \in \mathbb{Q}^m$ a non-negative vector.

$\Longrightarrow$ Suppose that a LIF $f(\mathbf{x}) = \mathbf{d^T} \cdot \mathbf{x}$ exists for P. Then by Definition 2, we have $\forall \mathbf{x}, \mathbf{x}'.(\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c} \Rightarrow f(\mathbf{x}') - f(\mathbf{x}) \geq \delta > 0$. That is to say $\forall \mathbf{x}, \mathbf{x}'.(\mathbf{H}\mathbf{x} + \mathbf{H}'\mathbf{x}') \leq \mathbf{c} \Rightarrow \mathbf{d^T} \cdot \mathbf{x}' - \mathbf{d^T} \cdot \mathbf{x} \geq \delta > 0$. By Farkas' lemma, this is equivalent to $\exists \mu \geq 0, \mathbf{d^T} = -\mu\mathbf{H}' \wedge -\mathbf{d^T} = -\mu\mathbf{H} \wedge -\delta \geq \mu\mathbf{c}$, i.e., $\exists \mu \geq 0$, s.t. $\mu(\mathbf{H}' + \mathbf{H}) = 0 \wedge \mu\mathbf{c} < 0$.

$\Longleftarrow$ Define $\exists \mu \geq 0$, s.t. $\mu(\mathbf{H} + \mathbf{H}') = 0 \wedge \mu\mathbf{c} < 0$. We predefine a linear function template by $f(\mathbf{x}) = \mathbf{d^T} \cdot \mathbf{x}$.

Since $\exists \mu \geq 0$, s.t. $\mu(\mathbf{H} + \mathbf{H}') = 0 \wedge \mu\mathbf{c} < 0$, and $(\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$, we have

$$\begin{aligned} \mu(\mathbf{H}\mathbf{x} + \mathbf{H}'\mathbf{x}') &\leq \mu\mathbf{c}, \\ \mu\mathbf{H}\mathbf{x} + \mu\mathbf{H}'\mathbf{x}' &\leq \mu\mathbf{c}, \\ -\mu\mathbf{H}'\mathbf{x}' - \mu\mathbf{H}\mathbf{x} &\geq -\mu\mathbf{c}, \\ -\mu\mathbf{H}'\mathbf{x}' - (-\mu\mathbf{H}')\mathbf{x} &\geq -\mu\mathbf{c} > 0. \end{aligned}$$

After simplification, we find that $-\mu\mathbf{H}'\mathbf{x}' - (-\mu\mathbf{H}')\mathbf{x} \geq -\mu\mathbf{c} > 0$ can be written as $f(x') - f(x) \geq \delta > 0$, where $f(\mathbf{x}) = \mathbf{d^T} \cdot \mathbf{x}$, $\mathbf{d^T} = -\mu\mathbf{H}'$ and $\delta = -\mu\mathbf{c}$. since $\forall \mathbf{x}, \mathbf{x}'.((\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c} \Rightarrow f(\mathbf{x}') - f(\mathbf{x}) \geq \delta > 0)$, by Definition 2, $f(\mathbf{x})$ is a LIF for P. This completes the proof of the theorem. ☐

Remark 3. *By Theorem 2, if System (7) is satisfiable, then we can construct a LIF $f(\mathbf{x})$=$\mathbf{d^T} \cdot \mathbf{x} = -\mu\mathbf{H}'\mathbf{x}$, where μ is a non-negative vector over the rationals.*

Given a single-path linear constraint loop program $P : (\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$, let $f(\mathbf{x}) = \mathbf{d^T} \cdot \mathbf{x}$ be a LIF. By adding $f(\mathbf{x}) \geq k$ into the system corresponding to P, one can construct a new system $(\mathbf{N}, \mathbf{N}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{e}$, which corresponding to a new linear loop P', having loop guard $\mathbf{B}\mathbf{x} \leq \mathbf{b} \wedge f(\mathbf{x}) \geq k$. For the new system, $\mathbf{N} = \begin{pmatrix} \mathbf{H} \\ -\mathbf{d^T} \end{pmatrix} \in \mathbb{Q}^{(m+1) \times n}$, $\mathbf{N}' = \begin{pmatrix} \mathbf{H}' \\ \mathbf{0} \end{pmatrix} \in \mathbb{Q}^{(m+1) \times n}$ and $\mathbf{e} = \begin{pmatrix} \mathbf{c} \\ -k \end{pmatrix} \in \mathbb{Q}^{m+1}$.

Theorem 3. ***(Test Existence of Eventual Linear Ranking Functions)*** *Given a single-path linear constraint loop program $P : (\mathbf{H}, \mathbf{H}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}$. An eventual linear ranking function over Ω exists iff there exist non-negative vectors over the rationals λ_1 and λ_2 such that the following system is satisfiable.*

$$S_1 : \begin{array}{r} \mu \geq 0, \\ \mu(\mathbf{H} + \mathbf{H}') = 0, \\ \mu\mathbf{c} < 0, \end{array} \quad S_\rho : \begin{array}{r} \lambda_1, \lambda_2 \geq 0, \\ \lambda_1\mathbf{N}' = 0, \\ (\lambda_1 - \lambda_2)\mathbf{N} = 0, \\ \lambda_2(\mathbf{N} + \mathbf{N}') = 0, \\ \lambda_2\mathbf{e} < 0. \end{array} \tag{8}$$

$$\mu, \lambda_1, \lambda_2 \geq 0, \tag{9a}$$
$$\mu(\mathbf{H} + \mathbf{H}') = 0, \tag{9b}$$
$$\mu\mathbf{c} < 0, \tag{9c}$$
$$\lambda_1\mathbf{N}' = 0, \tag{9d}$$
$$(\lambda_1 - \lambda_2)\mathbf{N} = 0, \tag{9e}$$
$$\lambda_2(\mathbf{N} + \mathbf{N}') = 0, \tag{9f}$$
$$\lambda_2\mathbf{e} < 0. \tag{9g}$$

Proof. By Remark 2, we know that if there exists an eventual linear ranking function $\rho(\mathbf{x})$ over Ω, then there exists a LIF $f(\mathbf{x})$, such that $\rho(\mathbf{x})$ is a linear ranking function over $\Omega\cap\Omega_f$, which is the region characterized by loop guard of P'. Clearly, by Theorem 1,it follows that P' has linear ranking functions over $\Omega\cap\Omega_f$ iff System (8) is satisfiable. This completes the proof of the theorem. □

Remark 4. *Given λ_1 and λ_2, solutions of System (8), a linear ranking function is defined by $\rho(\mathbf{x}) = \mathbf{r}^T\cdot\mathbf{x}$, where $\mathbf{r} = \lambda_2\mathbf{N}'$.*

Example 2. *Following Example 1, by the above definitions, P_1 can be written as*

$$\begin{pmatrix}-1 & 0 & 0 & 0\\ -1 & -1 & 1 & 0\\ 0 & -1 & 0 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_1'\\ x_2'\end{pmatrix} \le \begin{pmatrix}0\\ 0\\ -1\end{pmatrix},$$

where: $\mathbf{H} = \begin{pmatrix}-1 & 0\\ -1 & -1\\ 0 & -1\end{pmatrix}$, $\mathbf{H}' = \begin{pmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{pmatrix}$, $\mathbf{x}=(x_1, x_2)^T$, $\mathbf{x}'=(x_1', x_2')^T$, *and* $\mathbf{c} = (0, 0, -1)^T$.

Synthesizing Linear Increasing Functions for P_1

Define $\mu = (\mu_{11}, \mu_{12}, \mu_{13})$. Let $f(\mathbf{x}) = \mathbf{d^T x} = d_1x_1+d_2x_2$ be a LIF template. By Theorem 2, LIF exist iff there exists an non-negative vector μ such that the following system is satisfiable.

$$S_1 : \begin{cases} \mu & \ge & 0\\ \mu(\mathbf{H}+\mathbf{H}') & = & 0\\ \mu\mathbf{c} & < & 0.\end{cases}$$

By simplification, we have $\{-\mu_{11} = 0, -\mu_{12} = 0, -\mu_{13} < 0\}$ and $\mathbf{d} = -\mu A' = (-\mu_{12}, -\mu_{13})$. Therefore, we can get the space of LIF $\Psi = \{(-\mu_{12}, -\mu_{13})| - \mu_{11} = 0, -\mu_{12} = 0, \mu_{13} > 0\}$.

Synthesizing Eventual Linear Ranking Functions for P_1

Predefine $\rho(\mathbf{x}) = \mathbf{r}^T\cdot\mathbf{x} = r_1x_1 + r_2x_2$ as an eventual LRF template. According to Definition 3, consider the following formula,

$$\begin{aligned}\exists\mathbf{r}, k_1.\forall\mathbf{x},\mathbf{x}' : \big(\tau(\mathbf{x},\mathbf{x}')\wedge f(\mathbf{x})\ge k_1\big) \Rightarrow\\ \big(\rho(\mathbf{x})\ge 0\wedge\rho(\mathbf{x})-\rho(\mathbf{x}')\ge 1\big).\end{aligned}\tag{10}$$

The implication antecedent for System (10), $\tau(\mathbf{x},\mathbf{x}')$ and $f(\mathbf{x})\ge k_1$, can be written as the system $(\mathbf{N}, \mathbf{N}')\begin{pmatrix}\mathbf{x}\\ \mathbf{x}'\end{pmatrix}\le\mathbf{e}$, *where* $\mathbf{N} = \begin{pmatrix}-1 & 0\\ -1 & -1\\ 0 & -1\\ \mu_{12} & \mu_{13}\end{pmatrix}$, $\mathbf{N}'=\begin{pmatrix}0 & 0\\ 1 & 0\\ 0 & 1\\ 0 & 0\end{pmatrix}$, $\mathbf{e}=\begin{pmatrix}0\\ 0\\ -1\\ -k_1\end{pmatrix}$.

Let $\lambda_1 = (\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14})$ and $\lambda_2 = (\lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{24})$. By Theorem 3, Formula (10) can be equivalently written as the following system:

$$S_\rho : \begin{cases}\lambda_1,\lambda_2 & \ge & 0\\ \lambda_1\mathbf{N}' & = & 0\\ (\lambda_1-\lambda_2)\mathbf{N} & = & 0\\ \lambda_2(\mathbf{N}+\mathbf{N}') & = & 0\\ \lambda_2\mathbf{e} & < & 0.\end{cases}\tag{11}$$

By Remark 4, we know that $\rho(\mathbf{x}) = \mathbf{r}\cdot\mathbf{x} = \lambda_2\mathbf{N}'\mathbf{x} = (\lambda_{22}, \lambda_{23})\mathbf{x}$. Thus, we need to get the satisfiable λ_2 by eliminating quantified μ and λ_1 from System (11).

For eliminating quantifiers, we introduce the tool RegularChains, which is a package in the computer algebra software Maple. In the following, we directly use RegularChains to eliminate μ, λ_1 from System (11).

1: with(RegularChains): with(SemiAlgebraicSetTools);
$2 : R_1 = \&E([\mu_{11}, \mu_{12}, \mu_{13}, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14}, \lambda_{21}, \lambda_{24}, k_1]), (\mu_{11}\ge 0)\&and(\mu_{12}\ge 0)\&and(\mu_{13}\ge 0)\&and(\lambda_{11}\ge 0)\&and(\lambda_{12}\ge 0)\&and(\lambda_{13}\ge 0)\&and(\lambda_{14}\ge 0)\&and(\lambda_{21}\ge 0)\&and(\lambda_{22}\ge 0)\&and(\lambda_{23}\ge 0)\&and(\lambda_{24}\ge 0)\&and(-\mu_{11} = 0)\&and(-\mu_{12} = 0)\&and(-\mu_{13} < 0)\&and(\lambda_{12} = 0)\&and(\lambda_{13} = 0)\&and(-(\lambda_{11}-\lambda_{21}) - (\lambda_{12}-\lambda_{22}) + \mu_{12}(\lambda_{14}-\lambda_{24}) = 0)\&and(-(\lambda_{12}-\lambda_{22}) - (\lambda_{13}-\lambda_{23}) + \mu_{13}(\lambda_{14}-\lambda_{24}) = 0)\&and(-\lambda_{21}+\mu_{12}\lambda_{24} = 0)\&and(-\lambda_{22}+\mu_{13}\lambda_{24} = 0)\&and(-\lambda_{23}-\lambda_{24}k_1 < 0);$
3: QuantifierElimination(R_1,output=rootof);

Executing the above commands, we obtain $\{\lambda_{22} > 0, \lambda_{23} = 0\}$. Thus, the desired eventual linear ranking function exists, which implies that the loop program P_1 is terminating by Theorem 3.

For loop Program P_1, if we use the method in [3], which is based on Farkas' lemma, then the involved system contains 14 variables. But the system derived by Theorem 3 contains 12 variables. This can be seen from $\mathbb{S}_2$ or System 11. Thus, to some extend, Theorem 3 can decrease the complexity of computation of eventual linear ranking functions.

Example 3. *Consider the following single-path linear constraint loop P_2:*

while $(x_1-x_3\ge 0)$ $\{x_1'\le x_1+x_2; x_2'\le x_2-x_3; x_3'\ge x_3+1;\}$

In this example, there does not exist a linear ranking function. Furthermore, we fail to synthesize the eventual linear ranking functions for P_2 by Theorem 3.

In the next section, we will give the definition of $l-$depth eventual ranking functions. And we will show that for given an SLC loop, the existence of $l-$depth eventual ranking functions implies the loop is terminating.

IV. The L-depth Eventual Linear Ranking Functions for SLC Loops

In this section, we present a method to detect the existence of $l-$depth eventual LRFs for further termination analysis.

By Remark 2, we know for a single-path linear constraint loop, an eventual LRF over Ω is a LRF over $\Omega\wedge\Omega_f$ where $\Omega_f = \{\mathbf{x}\in\mathbb{Q}^n | f(\mathbf{x})\ge k\}$. However, for some single-path linear constraint loops, LRFs over $\Omega\cap\Omega_f$ may not exist, we need to further divide the region $\Omega\cap\Omega_f$ by adding new LIFs. First, we give the definition of $l-$depth LIFs.

Let $f_1(\mathbf{x})$ be a LIF over Ω. We say that $f_1(\mathbf{x})$ is a 2-depth LIF over Ω if it satisfies

$$\exists k_1, \forall \mathbf{x}, \mathbf{x}', ((\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1) \Rightarrow f_2(\mathbf{x}') - f_2(\mathbf{x}) \geq 1). \quad (12)$$

Generally, we give the following definition of $l-$depth LIFs over Ω.

Definition 4. ***(L-depth Linear Increasing Functions)*** *A linear function $f_l(\mathbf{x}) = \mathbf{d}_l^T\mathbf{x}$ is an $l-$depth linear increasing function over Ω if the following formula holds,*

$$\exists k_1, k_2, ..., k_{l-1}, \forall \mathbf{x}, \mathbf{x}', ((\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge \\ ... \wedge f_{l-1}(\mathbf{x}) \geq k_{l-1}) \Rightarrow f_l(\mathbf{x}') - f_l(\mathbf{x}) \geq 1) \quad (13)$$

In the above formula, $f_1(\mathbf{x}), ..., f_{l-1}(\mathbf{x})$ are $1-$depth,...,$(l-1)-$depth LIFs over Ω. For some k_i, define $\Omega_{f_i} = \{\mathbf{x} \in \mathbb{Q}^n | f_i(\mathbf{x}) \geq k_i\}, i \in [1, l]$.

Remark 5. *It is not difficult to see that if f_l is an $l-$depth LIF over Ω for P, then f_l is a LIF over $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{l-1}}$ for P'. Then P' can be written as*

while$(\mathbf{Bx} \leq \mathbf{b} \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_{l-1}(\mathbf{x}) \geq k_{l-1})$ *do* $\mathbf{x}' \leq \mathbf{Ax} + \mathbf{v}$.

Based on the form of Formula (2), P' can be translated into $(\mathbf{M}, \mathbf{M}')\begin{pmatrix}\mathbf{x}\\ \mathbf{x}'\end{pmatrix} \leq \mathbf{c}_l$, where $\mathbf{M} = (\mathbf{B}, -\mathbf{A}, -\mathbf{d}_1^T, ..., -\mathbf{d}_{l-1}^T)^T = (\mathbf{H}, -\mathbf{d}_1^T, ..., -\mathbf{d}_{l-1}^T)^T \in \mathbb{Q}^{(p+n+l-1)\times n}$, $\mathbf{M}' = (\mathbf{0}, \mathbf{I}, \mathbf{0}, ..., \mathbf{0})^T = (\mathbf{H}', \mathbf{0}, ..., \mathbf{0})^T \in \mathbb{Q}^{(p+n+l-1)\times n}$, $\mathbf{c}_l = (\mathbf{b}, \mathbf{v}, -k_1, ..., -k_{l-1})^T = (\mathbf{c}, -k_1, ..., -k_{l-1})^T \in \mathbb{Q}^{p+n+l-1}$.

Clearly, when $l = 1$, the $l-$depth LIF is exactly LIF as defined in Definition 2.

Let Ψ_l be the space of $l-$depth LIFs $f_l(\mathbf{x}) = \mathbf{d}_l \cdot \mathbf{x}$ over Ω, $\Psi_l = \{\mathbf{d}_l \in \mathbb{Q}^n | \exists k_1, ..., k_{l-1}, \forall \mathbf{x}, \mathbf{x}', (\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_{l-1}(\mathbf{x}) \geq k_{l-1} \Rightarrow f_l(\mathbf{x}') - f_l(\mathbf{x}) \geq 1)\}$. For some k_i, set $\Omega_{f_i} = \{\mathbf{x} \in \mathbb{Q}^n | f_i(\mathbf{x}) \geq k_i\}$. And a simple result can be obtained as follows.

Theorem 4. ***(Relations among the Space of Linear Increasing Functions)*** *Given an SLC loop P, let $f_l(\mathbf{x})$ be an $l-$depth linear increasing function over Ω and let $f_{l+1}(\mathbf{x})$ be an $(l+1)-$depth linear increasing function over Ω. Then, $\Psi_l \subseteq \Psi_{l+1}$.*

Example 4. *For the SLC loop program P_2 in Example 3, P_2 can be written as $(\mathbf{H}, \mathbf{H}')\begin{pmatrix}\mathbf{x}\\ \mathbf{x}'\end{pmatrix} \leq \mathbf{c}_0$, where* $\mathbf{H} = \begin{pmatrix}-1 & 0 & 1\\ -1 & -1 & 0\\ 0 & -1 & 1\\ 0 & 0 & 1\end{pmatrix}$, $\mathbf{H}' = \begin{pmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{pmatrix}$, $\mathbf{c}_0 = (-1, 0, 0, -1)^T$.
Next, we illustrate how to synthesize a 2-depth LIF.

Let $f_1(\mathbf{x}) = \mathbf{d_1^T}\mathbf{x} = d_{11}x_1 + d_{12}x_2 + d_{13}x_3$ be a $1-$depth LIF template and let $f_2(\mathbf{x}) = \mathbf{d_2^T}\mathbf{x} = d_{21}x_1 + d_{22}x_2 + d_{23}x_3$ be a $2-$depth LIF template. By Definition 4, the following formula should be satisfied.

$$\exists k_1. \forall \mathbf{x}, \mathbf{x}' : (\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1) \Rightarrow f_2(\mathbf{x}') - f_2(\mathbf{x}) \geq 1).$$

To synthesize $f_2(\mathbf{x})$, we need to compute the $1-$depth linear ranking function $f_1(\mathbf{x})$ first. By the argument mentioned above, when $l = 1$, the definition of 1-depth LIF is the same with Definition 2. By Theorem 2, to compute $f_1(\mathbf{x})$ is equivalent to check if the following formula is satisfiable,

$$S_1 : \begin{cases} \mu_1 & \geq & 0\\ \mu_1(\mathbf{H} + \mathbf{H}') & = & 0\\ \mu_1\mathbf{c}_0 & < & 0, \end{cases} \quad (14)$$

where $\mu_1 = (\mu_{11}, \mu_{12}, \mu_{13}, \mu_{14})$. System (14) can be written as $\{\mu_{11} \geq 0, \mu_{12} \geq 0, \mu_{13} \geq 0, \mu_{14} \geq 0, -\mu_{11} = 0, -\mu_{12} = 0, \mu_{11} + \mu_{13} = 0, -\mu_{14} < 0\}$. By Remark 3, $f_1(x) = -\mu_1\mathbf{H}'\mathbf{x} = (-\mu_{12}, -\mu_{13}, \mu_{14})(x_1, x_2, x_3)^T$. Therefore, we can get $\Psi_1 = \{(d_{11}, d_{12}, d_{13})\} = \{(-\mu_{12}, -\mu_{13}, \mu_{14}) | \mu_{11} \geq 0, \mu_{12} \geq 0, \mu_{13} \geq 0, \mu_{14} \geq 0, -\mu_{11} = 0, -\mu_{12} = 0, \mu_{11} + \mu_{13} = 0, \mu_{14} > 0\}$, which is the space of 1-depth LIF for P_2.

Next,we will synthesize $f_2(\mathbf{x})$. Based on loop program P_2, Ω=$\{\mathbf{x} \in \mathbb{Q}^n | x_1 - x_3 \geq 0\}$ and Ω_{f_1}=$\{\mathbf{x} \in \mathbb{Q}^n | f_1(\mathbf{x}) \geq k_1\}$. By Remark 5, we can construct a new loop P_2',

while $(x_1 - x_3 \geq 0 \wedge f_1(\mathbf{x}) \geq k_1)$
$\{x_1' \leq x_1 + x_2; x_2' \leq x_2 - x_3; x_3' \geq x_3 + 1;\}$.

And P_2' can be translated into $(\mathbf{M}, \mathbf{M}')\begin{pmatrix}\mathbf{x}\\ \mathbf{x}'\end{pmatrix} \leq \mathbf{c}_2$, where $\mathbf{M} = \begin{pmatrix}\mathbf{H}\\ -\mathbf{d_1^T}\end{pmatrix}$, $\mathbf{M}' = \begin{pmatrix}\mathbf{H}\\ \mathbf{0}\end{pmatrix}$ and $\mathbf{c_1} = \begin{pmatrix}\mathbf{c}_0\\ -k_1\end{pmatrix}$. Then we can get 2-depth LIFs by solving the following systems,

$$S_2 : \begin{cases} \mu_2 & \geq & 0\\ \mu_2(\mathbf{M} + \mathbf{M}') & = & 0\\ \mu_2\mathbf{c_1} & < & 0, \end{cases} \quad (15)$$

where $\mu_2 = (\mu_{21}, \mu_{22}, \mu_{23}, \mu_{24}, \mu_{25})$.

Clearly, Formula (15) can be written as $\{\mu_{21} \geq 0, \mu_{22} \geq 0, \mu_{23} \geq 0, \mu_{24} \geq 0, \mu_{25} \geq 0, -\mu_{21} + \mu_{12}\mu_{25} = 0, -\mu_{22} + \mu_{13}\mu_{25} = 0, \mu_{21} + \mu_{23} - \mu_{14}\mu_{25} = 0, -\mu_{24} - k_1\mu_{25} < 0\}$. By Theorem 2, the 2-depth LIF is $f_2(\mathbf{x}) = -\mu_2\mathbf{M}'\mathbf{x} = (-\mu_{22}, -\mu_{23}, \mu_{24})(x_1, x_2, x_3)^T$. Therefore, the space of the 2-depth LIF is $\Psi_2 = \{(d_{21}, d_{22}, d_{23})\} = \{(-\mu_{22}, -\mu_{23}, \mu_{24}) | \mu_{11} \geq 0, \mu_{12} \geq 0, \mu_{13} \geq 0, \mu_{14} \geq 0, -\mu_{11} = 0, -\mu_{12} = 0, \mu_{11} + \mu_{13} = 0, -\mu_{14} < 0, \mu_{21} \geq 0, \mu_{22} \geq 0, \mu_{23} \geq 0, \mu_{24} \geq 0, \mu_{25} \geq 0, -\mu_{21} + \mu_{12}\mu_{25} = 0, -\mu_{22} + \mu_{13}\mu_{25} = 0, \mu_{21} + \mu_{23} - \mu_{14}\mu_{25} = 0, -\mu_{24} - k_1\mu_{25} < 0\}$.

Next, we introduce the definition of $l-$depth eventual linear ranking functions for single-path linear constraint Loops.

Definition 5. ***(L-depth Eventual LRFs for SLC Loops)*** *Given an SLC loop P, let $f_1(\mathbf{x}), ..., f_l(\mathbf{x})$ be the 1-depth,...,l-depth linear increasing function over Ω. A linear function $\rho(\mathbf{x})$ is an l-depth eventual linear ranking function over Ω if the following formula is satisfiable.*

$$\exists k_1, ..., k_l, \forall \mathbf{x}, \mathbf{x}', \big((\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge \\ f_l(\mathbf{x}) \geq k_l) \Rightarrow \rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1\big) \quad (16)$$

Remark 6. *If $\rho(\mathbf{x})$ is an l-depth eventual linear ranking function over Ω for P. Then it is easy to see that $\rho(\mathbf{x})$ can be* *regarded as a linear ranking function over $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$ for P'. Then P' can be written as*

$$\textit{while}(\mathbf{B}\mathbf{x} \leq \mathbf{b} \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_l(\mathbf{x}) \geq k_l) \textit{ do } \mathbf{x}' \leq \mathbf{A}\mathbf{x} + \mathbf{v}.$$

Based on the form of Formula (2), P' can be translated into $(\mathbf{N}, \mathbf{N}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{e}_l$, where $\mathbf{N} = (\mathbf{B}, -\mathbf{A}, -\mathbf{d}_1^T, ..., -\mathbf{d}_l^T)^T = (\mathbf{H}, -\mathbf{d}_1^T, ..., -\mathbf{d}_l^T)^T \in \mathbb{Q}^{(p+n+l)\times n}$, $\mathbf{N}' = (\mathbf{0}, \mathbf{I}, \mathbf{0}, ..., \mathbf{0})^T = (\mathbf{H}', \mathbf{0}, ..., \mathbf{0})^T \in \mathbb{Q}^{(p+n+l)\times n}$, $\mathbf{e}_l = (\mathbf{b}, \mathbf{v}, -k_1, ..., -k_l)^T = (\mathbf{c}, -k_1, ..., -k_l)^T \in \mathbb{Q}^{p+n+l}$.

Proposition 1. *Given an SLC loop P, let $f_1(\mathbf{x}), ..., f_{i-1}(\mathbf{x})$ be 1-depth,...,$(i-1)$-depth LIFs over Ω and suppose $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}} \neq \varnothing$. Then, for any $\mathbf{x} \in \Omega - \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$ after a finite number of iterations, and for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$, after an arbitrary number of iterations, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}} \cup ... \cup \overline{\Omega_{f_{i-1}}}$.*

Proof. For convenience, we just consider two cases when $i = 2$ and $i = 3$, the other cases can be analyzed using similar method.

When $i = 2$, by hypothesis, $\Omega \cap \Omega_{f_1} \neq \varnothing$. Clearly, $\Omega = (\Omega - \Omega_{f_1}) \cup (\Omega \cap \Omega_{f_1})$. Namely, Ω can be divided into two mutually exclusive parts by $f_1(\mathbf{x}) = k_1$, $\Omega - \Omega_{f_1}$ and $\Omega \cap \Omega_{f_1}$.

Since $f_1(\mathbf{x})$ are 1-depth LIFs over Ω, by Remark 1, we know that for any $\mathbf{x} \in \Omega - \Omega_{f_1}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1}$ after a finite number of iterations and for any $\mathbf{x} \in \Omega \cap \Omega_{f_1}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}}$ after an arbitrary number of iterations.

When $i = 3$, we know that $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2} \neq \varnothing$ by hypothesis. Obviously, $\Omega = (\Omega - \Omega_{f_1} \cap \Omega_{f_2}) \cup (\Omega \cap \Omega_{f_1} \cap \Omega_{f_2})$.

$\Omega - \Omega_{f_1} \cap \Omega_{f_2} = (\Omega - \Omega_{f_1}) \cup (\Omega \cap \Omega_{f_1} - \Omega_{f_2})$. For any $\mathbf{x} \in \Omega - \Omega_{f_1}$, by the above proof, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1}$ after a finite number of iterations. $\Omega \cap \Omega_{f_1}$ can be divided into two parts by $f_2(\mathbf{x}) = k_2$, therefore, $\Omega \cap \Omega_{f_1} = (\Omega \cap \Omega_{f_1} - \Omega_{f_2}) \cup (\Omega \cap \Omega_{f_1} \cap \Omega_{f_2})$. We next proof that for any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) - \Omega_{f_2}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2}$ after a finite number of iterations.

For any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) - \Omega_{f_2}$, by Remark 1, $\mathbf{x}$ must fall into $\overline{\Omega \cap \Omega_{f_1}}$ or $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2}$ after a finite number of iterations since $f_2(\mathbf{x})$ is a LIF over $\Omega \cap \Omega_{f_1}$. Because $\overline{\Omega \cap \Omega_{f_1}} = \overline{\Omega} \cup \overline{\Omega_{f_1}}$, we next will further claim that $\forall \mathbf{x} \in (\Omega \cap \Omega_{f_1}) - \Omega_{f_2}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}}$ after an arbitrary number of iterations. Since $f_1(\mathbf{x})$ is 1-depth LIF over Ω, by Remark 1, for any $\mathbf{x} \in \Omega \cap \Omega_{f_1}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}}$. Hence, for any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) - \Omega_{f_2}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}}$ after an arbitrary number of iterations, since $(\Omega \cap \Omega_{f_1}) - \Omega_{f_2} \subseteq (\Omega \cap \Omega_{f_1})$. Thus, for any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) - \Omega_{f_2}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2}$ after a finite number of iterations. Therefore, for any $\mathbf{x} \in \Omega - \Omega_{f_1} \cap \Omega_{f_2}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2}$ after a finite number of iterations.

For any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) \cap \Omega_{f_2}$, by Remark 1, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_2}}$ after an arbitrary number of iterations since $f_2(\mathbf{x})$ is a LIF over $\Omega \cap \Omega_{f_1}$. In addition, because $f_1(\mathbf{x})$ is 1-depth LIF over Ω, for any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) \cap \Omega_{f_2} \subseteq \Omega \cap \Omega_{f_1}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}}$ after an arbitrary number of iterations. Therefore, for any $\mathbf{x} \in (\Omega \cap \Omega_{f_1}) \cap \Omega_{f_2}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}} \cup \overline{\Omega_{f_2}}$ after an arbitrary number of iterations.

Analogously, since $f_1(\mathbf{x}), ..., f_{i-1}(\mathbf{x})$ are 1-depth,...,$(i-1)$-depth LIFs over Ω and $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}} \neq \varnothing$. Using the similar arguments mentioned above, we get for any $\mathbf{x} \in \Omega - \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$ after an arbitrary number of iterations. Meanwhile, for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$, after a finite number of iterations, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}} \cup ... \cup \overline{\Omega_{f_{i-1}}}$. □

Theorem 5. *Given a single-path linear constraint loop program P, if there is an l-depth eventual linear ranking function for P, then P always terminates.*

Proof. First, we recall some notations. $\tau = \{\mathbf{B}\mathbf{x} \leq \mathbf{b} \wedge \mathbf{x}' \leq \mathbf{A}\mathbf{x} + \mathbf{v}\}$, $\Omega = \{\mathbf{x} \in \mathbb{Q}^n | \mathbf{B}\mathbf{x} \leq \mathbf{b}\}$.

If there is an l-depth eventual linear ranking function for P, say $\rho(\mathbf{x})$, then by Definition 5, we know that Formula (16) is true. That is $\exists k_1, ..., k_l$, s.t.,

$$\forall \mathbf{x}, \mathbf{x}', \big((\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_l(\mathbf{x}) \geq k_l) \Rightarrow \rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1\big) \tag{17}$$

is true, where $f_1(\mathbf{x}), ..., f_{i-1}(\mathbf{x})$ are 1-depth,...,$(i-1)$-depth LIFs over Ω. Let $\Omega_{f_i} = \{\mathbf{x} \in \mathbb{Q}^n | f_i(\mathbf{x}) \geq k_i\}$. Since Formula (17) is true, there are two cases to consider.

Case 1. When the system $(\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_l(\mathbf{x}) \geq k_l)$ has no solution. It is not difficult to see that this is equivalent to $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l} = \varnothing$. Hence, $\exists i \in [1, l]$, s.t., $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}} \neq \varnothing \wedge \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}} \cap \Omega_{f_i} = \varnothing$. Set $\Omega' = \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$. Because $\Omega = (\Omega - \Omega') \cup \Omega'$, we will show that for any $\mathbf{x} \in (\Omega - \Omega') \cup \Omega'$ $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations.

1) First, let us consider the point in $\Omega - \Omega'$. Since $f_1(\mathbf{x})$,..., $f_{i-1}(\mathbf{x})$ are 1-depth,...,(i-1)-depth LIFs over Ω and $\Omega' \neq \varnothing$, by proposition 1, for any $\mathbf{x} \in \Omega \cap \overline{\Omega'}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ or Ω' after a finite number of iterations.
2) Second, we will further show that for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations. Since $\Omega' \cap \Omega_{f_i} = \varnothing$, for any $\mathbf{x} \in \Omega'$, we have $f_i(\mathbf{x}) < k_i$. Suppose that there exists an infinite iterative sequence $S = \{\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, ...\} \subseteq \Omega'$. Then there exists $\mathbf{x}_N \in S$, s.t. $f_i(\mathbf{x}_N) \geq k_i$ because $f_i(\mathbf{x})$ is a LIF over Ω' by Remark 5. This contradicts the hypothesis that for any $\mathbf{x} \in \Omega'$, $f_i(\mathbf{x}) < k_i$. Therefore, the infinite sequence S does not exist in Ω'. That is to say, for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ must fall into $\overline{\Omega'}$ after a finite number of iterations. Furthermore, since $f_1(\mathbf{x})$,..., $f_{i-1}(\mathbf{x})$ are 1-depth,...,(i-1)-depth LIFs over Ω and $\Omega' \neq \varnothing$, by Proposition 1, for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1}} \cup ... \cup \overline{\Omega_{f_{i-1}}}$ after an arbitrary number of iterations. That is to say, for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ must fall into $\overline{\overline{\Omega_{f_1}} \cup ... \cup \overline{\Omega_{f_{i-1}}}} = \Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}$ after an arbitrary number of iterations. By the above argument in 2), for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ must fall into $\overline{\Omega'} \cap (\Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}) = (\overline{\Omega} \cup \overline{\Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}}) \cap (\Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}) = \overline{\Omega} \cap (\Omega_{f_1} \cap ... \cap \Omega_{f_{i-1}}) \subseteq \overline{\Omega}$ after a finite number of iterations. Therefore, for any $\mathbf{x} \in \Omega'$, $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations.

Hence, by 1) and 2), for any $\mathbf{x} \in (\Omega - \Omega') \cup \Omega' = \Omega$, $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations. This implies that the loop terminates.

Case 2. When the system $(\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_l(\mathbf{x}) \geq k_l)$ has solutions, that is $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l} \neq \varnothing$. Let $\Omega = (\Omega - \Omega_{f_1} \cap ... \cap \Omega_{f_l}) \cup (\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l})$.

First, for any $\mathbf{x} \in (\Omega - \Omega_{f_1} \cap ... \cap \Omega_{f_l})$, by Proposition 1, after a finite number of iterations, $\mathbf{x}$ must fall into $\overline{\Omega}$ or $(\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l})$, because $f_1(\mathbf{x})$,..., $f_l(\mathbf{x})$ are 1-depth,...,l-depth LIFs over Ω and $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l} \neq \varnothing$.

Second, for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$, since $\rho(\mathbf{x})$ is an l-depth eventual linear ranking function over Ω, by Remark 6, we know $\rho(\mathbf{x})$ is a linear ranking function over $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$. This indicates that for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$, after a finite number of iterations, $\mathbf{x}$ must fall into $\overline{\Omega} \cup \overline{\Omega_{f_1} \cap ... \cap \Omega_{f_l}}$, the complement of $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$. Also, by Proposition 1, we know that for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$, $\mathbf{x}$ cannot fall into $\overline{\Omega_{f_1} \cap ... \cap \Omega_{f_l}} = \overline{\Omega_{f_1}} \cup ... \cup \overline{\Omega_{f_l}}$ after an arbitrary number of iterations. Thus, for any $\mathbf{x} \in \Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_l}$, $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations. Therefore, for any $\mathbf{x} \in \Omega$, $\mathbf{x}$ must fall into $\overline{\Omega}$ after a finite number of iterations. This implies that the loop terminates.

This completes the proof of the theorem. □

We can directly synthesize the LIF f_l over $\Omega \cap \Omega_{f_1} \cap ... \cap \Omega_{f_{l-1}}$ by Definition 4. Next, we introduce two algorithms to detect the i-depth LIFs and the l-depth eventual LRFs respectively.

Algorithm ***INCF***. Existence of i-depth LIFs for SLC loops

---

Input: an SLC loop program $P(\tau_{i-1})$.
Output: Return i-depth LIFs $f_i(\mathbf{x})$ over Ω and a system S_i.

1. Transform $P(\tau_{i-1})$ into the following form:

$$(\mathbf{M}_{i-1}, \mathbf{M}'_{i-1}) \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{c}_{i-1}.$$

2. Construct the following system by Theorem 2,

$$S_i : \begin{cases} \mu_i & \geq & 0 \\ \mu_i(\mathbf{M}_{i-1} + \mathbf{M}'_{i-1}) & = & 0 \\ \mu_i \mathbf{c}_{i-1} & < & 0, \end{cases}$$

where μ_i is a vector of parameters.

3. Set $f_i(\mathbf{x}) = -\mu_i \mathbf{M'}_{i-1}\mathbf{x}$ (the $i-$depth LIFs over Ω).
4. **return** $(f_i(\mathbf{x}), S_i)$;

Algorithm ***l-ELRF***. Existence of l-depth eventual LRFs for SLC Loops

---

Input: an SLC loop program $P(\tau)$ and a depth l.
Output: Return true if there exists an l-depth eventual LRF $\rho(\mathbf{x})$ over Ω. Otherwise, return false.

1. $\tau_0 = \tau$;
2. Ψ_1= the space of LIFs over Ω;
3. **if** $\Psi_1 = \varnothing$ **then**
4.   **return false**;
5. **end if**
6. **for**$(i = 1; i \leq l; i++)$
7. $(f_i(\mathbf{x}), S_i) =$ ***INCF***$(P(\tau_{i-1}))$;
8. $\tau_i = \tau_{i-1} \wedge f_i(\mathbf{x}) \geq k_i$;
9. **end for**
10. Transform $P(\tau_l)$ into the form of $(\mathbf{N}_l, \mathbf{N}'_l) \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{e}_l$;
11. By Theorem 3, one can construct a system S_ρ,

$$S_\rho : \begin{cases} \lambda_1, \lambda_2 & \geq & 0 \\ \lambda_1 \mathbf{N}'_l & = & 0 \\ (\lambda_1 - \lambda_2)\mathbf{N}_l & = & 0 \\ \lambda_2(\mathbf{N}_l + \mathbf{N}'_l) & = & 0 \\ \lambda_2 \mathbf{e}_l & < & 0, \end{cases}$$

where λ_1 and λ_2 are two vectors of parameters.

12. **if** $S_1 \wedge ... \wedge S_l \wedge S_\rho$ is satisfiable **then**
13.   **return true**;
14. **else**
15.   **return false**;
16. **end if**

In the above algorithms, $P(\tau_i)$ is an SLC loop program, where $\tau_i = \{\mathbf{Bx} \leq \mathbf{b} \wedge \mathbf{x}' \leq \mathbf{Ax} + \mathbf{v} \wedge f_1(\mathbf{x}) \geq k_1 \wedge ... \wedge f_{i-1}(\mathbf{x}) \geq k_{i-1}\}$.

Remark 7. *By Definition 4, Ψ_{i+1} is based on the loop P and LIFs $f_1(\mathbf{x}), ..., f_i(\mathbf{x})$. Hence, if $\Psi_1 = \varnothing$, $f_1(\mathbf{x})$ does not exist and $f_i(\mathbf{x})$ cannot be constructed for $i > 1$. By Theorem 4, the space $\Psi_l \subseteq \Psi_{l+1}$. Hence if $\Psi_1 \neq \varnothing$, then for any $i \in [2, l], \Psi_i \neq \varnothing$.*

Example 5. *For the SLC loop program P_2 from Example 3, we know that it has no eventual linear ranking functions. In this example, for simplification, set l=2 and test the existence of the 2-depth eventual linear ranking function. From Example 4, the space of 2-depth LIF Ψ_2 was obtained, and $\Omega = \{\mathbf{x} \in \mathbb{Q}^n | x_1 - x_3 \geq 0\}$, $\Omega_{f_1} = \{\mathbf{x} \in \mathbb{Q}^n | f_1(\mathbf{x}) \geq k_1\}$, $\Omega_{f_2} = \{\mathbf{x} \in \mathbb{Q}^n | f_2(\mathbf{x}) \geq k_2\}$.*

Define $\rho(\mathbf{x}) = \mathbf{r}^T \cdot \mathbf{x}$ as a 2-depth eventual LRF template. Following Definition 5, we have

$$\begin{aligned} \exists k_1, k_2. \forall \mathbf{x}, \mathbf{x}' : & \big(\tau(\mathbf{x}, \mathbf{x}') \wedge f_1(\mathbf{x}) \geq k_1 \wedge f_2(\mathbf{x}) \geq k_2\big) \\ & \Rightarrow \big(\rho(\mathbf{x}) \geq 0 \wedge \rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1\big). \end{aligned} \tag{18}$$

By Remark 6, we can construct a new loop P''_2,

while $(x_1 - x_3 \geq 0 \wedge f_1(\mathbf{x}) \geq k_1 \wedge f_2(\mathbf{x}) \geq k_2)$
$\{x'_1 \leq x_1 + x_2; x'_2 \leq x_2 - x_3; x'_3 \geq x_3 + 1;\}$.

And P''_2 can be translated into $(\mathbf{N}, \mathbf{N}') \begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix} \leq \mathbf{e_2}$, where:

$$\mathbf{N} = \begin{pmatrix} \mathbf{H} \\ -\mathbf{d_1^T} \\ -\mathbf{d_2^T} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \\ \mu_{12} & \mu_{13} & -\mu_{14} \\ \mu_{22} & \mu_{23} & -\mu_{24} \end{pmatrix}, \ \mathbf{N}' = \begin{pmatrix} \mathbf{H}' \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} =$$

$$\begin{pmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}, \mathbf{e_2} = \begin{pmatrix}\mathbf{c}_0\\ -k_1\\ -k_2\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\\ -1\\ -k_1\\ -k_2\end{pmatrix}.$$

Let $\lambda_3 = (\lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{34}, \lambda_{35}, \lambda_{36})$ *and* $\lambda_4 = (\lambda_{41}, \lambda_{42}, \lambda_{43}, \lambda_{44}, \lambda_{45}, \lambda_{46})$*. By the Theorem 3, eventual linear ranking functions over* $\Omega \cap \Omega_{f_1} \cap \Omega_{f_2}$ *exist iff the following system is satisfiable,*

$$S_\rho : \begin{cases} \lambda_3, \lambda_4 & \geq & 0\\ \lambda_3 \mathbf{N}' & = & 0\\ (\lambda_3 - \lambda_4)\mathbf{N} & = & 0\\ \lambda_4(\mathbf{N} + \mathbf{N}') & = & 0\\ \lambda_4 \mathbf{e_2} & < & 0. \end{cases}$$

Using RegularChains to eliminate the quantified μ_1, μ_2*. Next we get the result* $\{\lambda_{32} = \lambda_{33} = \lambda_{34} = \lambda_{36} = \lambda_{41} = \lambda_{43} = \lambda_{45} = 0, \lambda_{31} = \lambda_{42}, \lambda_{31} > 0, \lambda_{42} > 0, \lambda_{46} > 0, \lambda_{35} \geq 0, \lambda_{44} \geq 0, \lambda_{44} \leq \lambda_{42}, k_1 > 0, k_2 > 0\}$.

By Remark 4, $\rho(\mathbf{x}) = \lambda_4 \mathbf{N}'\mathbf{x} = (\lambda_{42}, \lambda_{43}, -\lambda_{44})(x_1, x_2, x_3)^T$*, One particular solution is* $\lambda_{42} = \lambda_{44} = 1, \lambda_{43} = 0$*. Hence the 2-depth eventual linear ranking function is* $\rho(\mathbf{x}) = x_1 - x_3$*. Then the loop program* P_2 *is terminating by Theorem 5.*

V. Conclusion

In this paper, an algorithm has been proposed to synthesize l-depth eventual linear ranking functions for single-path linear constraint loops. Experiments show that for some loops having no traditional linear ranking functions defined in [3], [15], l-depth eventual linear ranking functions can be found. Besides, we extend the technique of Podelski and Rybalchenko in [22], to compute l-depth linear increasing functions and l-depth eventual linear ranking functions. Our algorithm is complete, i.e., for a given single-path linear constraint loop program, if an l-depth eventual linear ranking function indeed exists, then our algorithm can find it and return true. Otherwise, if such an l-depth eventual linear ranking function does not exist, our algorithm return false.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful suggestions. This research is supported by the National Natural Science Foundation of China NNSFC(61572024, 61103110, 11471307).

References

[1] C. Alias, A. Darte, P. Feautrier, et al. Multi-dimensional Rankings, Program Termination, and Complexity Bounds of Flowchart Programs[M]. Static Analysis. Springer Berlin Heidelberg. pages 117-133. 2010.

[2] T. Arts, J. Giesl. Termination of term rewriting using dependency pairs[J]. Theoretical Computer Science, 236(1):133-178. 2000.

[3] R. Bagnara and F. Mesnard. Eventual linear ranking functions. Proceedings of the 15th Symposium on Principles and Practice of Declarative Programming. Madrid, Spain, ACM. pages 229-238. 2013.

[4] R. Bagnara, F. Mesnard, A. Pescetti and E. Zaffanella.A new look at the automatic synthesis of linear ranking functions. Information and Computation, 215, pages 47-67. 2012.

[5] A. M. Ben-Amram.The Hardness of Finding Linear Ranking Functions for Lasso Programs. Electronic Proceedings in Theoretical Computer Science, 161, pages 32-45. 2014.

[6] A. M. Ben-Amram and S. Genaim. On the linear ranking problem for integer linear-constraint loops. POPL '13 Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages. Rome, Italy, ACM.pages 51-62. 2013.

[7] A. M. Ben-Amram and S. Genaim.Ranking Functions for Linear-Constraint Loops. Journal of the ACM 61(4), pages 1-55. 2014.

[8] A. Bradley, Z. Manna and H. Sipma. Linear Ranking with Reachability. Computer Aided Verification. K. Etessami and S. Rajamani, Springer Berlin Heidelberg. 3576: 491-504.2005.

[9] A. Bradley, Z. Manna and H. Sipma. The Polyranking Principle. Automata, Languages and Programming. L. Caires, G. Italiano, L. Monteiro, C. Palamidessi and M. Yung, Springer Berlin Heidelberg. 3580: 1349-1361.2005.

[10] A. Bradley, Z. Manna and H. Sipma. Termination Analysis of Integer Linear Loops. CONCUR 2005 C Concurrency Theory. M. Abadi and L. de Alfaro, Springer Berlin Heidelberg. 3653: 488-502.2005.

[11] M. Braverman. Termination of Integer Linear Programs. Computer Aided Verification. T. Ball and R. Jones, Springer Berlin Heidelberg. 4144: 372-385.2006.

[12] C. B. Chen, M. M Maza. Quantifier Elimination by Cylindrical Algebraic Decomposition Based on Regular Chains[C]. Proceedings of the 39th International Symposium on Symbolic and Algebraic ComputationACM, pp. 91-98. 2014.

[13] H. Y. Chen, S. Flur and S. Mukhopadhyay. Termination Proofs for Linear Simple Loops. Static Analysis. A. Miné and D. Schmidt, Springer Berlin Heidelberg. 7460: 422-438.2012.

[14] Y. H. Chen, B. C. Xia, L. Yang, N. J. Zhan and C. C. Zhou. Discovering Non-linear Ranking Functions by Solving Semi-algebraic Systems. Theoretical Aspects of Computing C ICTAC 2007. C. Jones, Z. Liu and J. Woodcock, Springer Berlin Heidelberg. 4711: 34-49.2007.

[15] M. Colón and H. Sipma. Synthesis of Linear Ranking Functions. Tools and Algorithms for the Construction and Analysis of Systems. T. Margaria and W. Yi, Springer Berlin Heidelberg. 2031: 67-81.2001.

[16] B. Cook, S. Gulwani, T. Lev-Ami, A. Rybalchenko and M. Sagiv. Proving Conditional Termination. Computer Aided Verification. A. Gupta and S. Malik, Springer Berlin Heidelberg. 5123: 328-340.2008.

[17] B. Cook, A. See and F. Zuleger. Ramsey vs. Lexicographic Termination Proving. Tools and Algorithms for the Construction and Analysis of Systems. N. Piterman and S. Smolka, Springer Berlin Heidelberg. 7795: 47-61.2013.

[18] P. Ganty and S. Genaim. Proving Termination Starting from the End. Computer Aided Verification. N. Sharygina and H. Veith, Springer Berlin Heidelberg. 8044: 397-412.2013.

[19] M. Heizmann, J. Hoenicke, J. Leike and A. Podelski. Linear Ranking for Linear Lasso Programs. Automated Technology for Verification and Analysis. D. Van Hung and M. Ogawa, Springer International Publishing. 8172: 365-380.2013.

[20] J. Leike and M. Heizmann. Ranking Templates for Linear Loops. Tools and Algorithms for the Construction and Analysis of Systems. E. brahm and K. Havelund, Springer Berlin Heidelberg. 8413: 172-186.2014.

[21] J. Leike and A. Tiwari. Synthesis for Polynomial Lasso Programs. Verification, Model Checking, and Abstract Interpretation. K. McMillan and X. Rival, Springer Berlin Heidelberg. 8318: 434-452.2014.

[22] A. Podelski and A. Rybalchenko. A Complete Method for the Synthesis of Linear Ranking Functions. Verification, Model Checking, and Abstract Interpretation. B. Steffen and G. Levi, Springer Berlin Heidelberg. 2937: 239-251.2004.

[23] K. Sohn, A. Van Gelder, Termination detection in logic programs using argument sizes (extended abstract), in: Proceedings of the Tenth ACM SIGACT- SIGMOD-SIGART Symposium on Principles of Database Systems, ACM, Association for Computing Machinery, Denver, CO, USA, 216C226, 1991.

[24] A. Tiwari. Termination of Linear Programs. Computer Aided Verification. R. Alur and D. Peled, Springer Berlin Heidelberg. 3114: 70-82.2004.