On Multiphase-Linear Ranking Functions

Amir M. Ben-Amram[1] and Samir Genaim[2(✉)]

[1] School of Computer Science, The Tel-Aviv Academic College, Tel Aviv, Israel
[2] DSIC, Complutense University of Madrid (UCM), Madrid, Spain
genaim@gmail.com

Abstract. Multiphase ranking functions ($M\Phi$RFs) were proposed as a means to prove the termination of a loop in which the computation progresses through a number of "phases", and the progress of each phase is described by a different linear ranking function. Our work provides new insights regarding such functions for loops described by a conjunction of linear constraints (single-path loops). We provide a complete polynomial-time solution to the problem of existence and of synthesis of $M\Phi$RF of bounded depth (number of phases), when variables range over rational or real numbers; a complete solution for the (harder) case that variables are integer, with a matching lower-bound proof, showing that the problem is coNP-complete; and a new theorem which bounds the number of iterations for loops with $M\Phi$RFs. Surprisingly, the bound is linear, even when the variables involved change in non-linear way. We also consider a type of *lexicographic* ranking functions more expressive than types of lexicographic functions for which complete solutions have been given so far. We prove that for the above type of loops, lexicographic functions can be reduced to $M\Phi$RFs, and thus the questions of complexity of detection, synthesis, and iteration bounds are also answered for this class.

1 Introduction

Proving that a program will not go into an infinite loop is one of the most fundamental tasks of program verification, and has been the subject of voluminous research. Perhaps the best known, and often used, technique for proving termination is the *ranking function*. This is a function f that maps program states into the elements of a well-founded ordered set, such that $f(s) > f(s')$ holds whenever state s' follows state s. This implies termination since infinite descent in a well-founded order is impossible.

Unlike termination of programs in general, which is the fundamental example of undecidability, the algorithmic problems of detection (deciding the existence) or generation (synthesis) of a ranking function can well be solvable, given certain choices of the program representation, and the class of ranking function.

This work was funded partially by the Spanish MINECO projects TIN2012-38137 and TIN2015-69175-C4-2-R, and by the CM project S2013/ICE-3006. We thank Mooly Sagiv for providing us with a working space at Tel-Aviv University, which was crucial for completing this work.

R. Majumdar and V. Kunčak (Eds.): CAV 2017, Part II, LNCS 10427, pp. 601–620, 2017.
DOI: 10.1007/978-3-319-63390-9_32

Numerous researchers have proposed such classes, with an eye towards decidability; in some cases the algorithmic problems have been completely settled, and efficient algorithms provided, while other cases remain as open problems. Thus, in designing ranking functions, we look for expressivity (to capture more program behaviors) but also want (efficient) computability. Besides proving termination, some classes of ranking functions also serve to bound the length of the computation (an *iteration bound*), useful in applications such as *cost analysis* (execution-time analysis, resource analysis) and loop optimization [1,2,7,14].

We focus on *single-path linear-constraint loops* (*SLC* loops for short), where a state is described by the values of a finite set of numerical variables, and the effect of a transition (one iteration of the loop) is described by a conjunction of *linear constraints*. We consider the setting of integer-valued variables, as well as rational-valued (or real-valued) variables[1]. Here is an example of this loop representation (a formal definition is in Sect. 2); primed variables $x', y', \ldots$ refer to the state following the transition.

$$\texttt{while}\ (x \geq -z)\ \texttt{do}\ x' = x + y,\ y' = y + z,\ z' = z - 1 \tag{1}$$

Note that by $x' = x + y$ we mean an equation, not an assignment statement; it is a standard procedure to compile sequential code into such equations (if the operations used are linear), or to approximate it using various techniques.

This constraint representation may be extended to represent branching in the loop body, a so-called *multiple-path loop*; in the current work we do not consider such loops. However, *SLC* loops are important, e.g., in approaches that reduce a question about a whole program to questions about simple loops [10,11,13,16,19]; see [21] for references that show the importance of such loops in other fields.

We assume the "constraint loop" to be given, and do not concern ourselves with the orthogonal topic of extracting such loops from general programs.

Types of Ranking Functions. Several types of ranking functions have been suggested; linear ranking functions (*LRFs*) are probably the most widely used and well-understood. In this case, we seek a function $f(x_1, \ldots, x_n) = a_1x_1 + \cdots + a_nx_n + a_0$, with the rationals as a co-domain, such that (i) $f(\bar{x}) \geq 0$ for any valuation $\bar{x}$ that satisfies the loop constraints (i.e., an enabled state); and (ii) $f(\bar{x}) - f(\bar{x}') \geq 1$ for any transition leading from $\bar{x}$ to $\bar{x}'$. Technically, the rationals are not a well-founded set under the usual order, but we can refer to the partial order $a \succeq b$ if and only if $a \geq 0$ and $a \geq b + 1$, which is well-founded. Given a linear-constraint loop, it is possible to find a linear ranking function (if one exists) using linear programming (*LP*). This method was found by multiple researchers in different contexts and in some alternative versions [9,14,22,24]. Since *LP* has a polynomial-time complexity, most of these methods yield polynomial-time algorithms. This method is sound (any ranking function produced is valid), and complete (if there is a ranking function, it will find one), when variables are

[1] For the results in this paper, the real-number case is equivalent to the rational-number case, and in the sequel we refer just to rationals.

assumed to range over the rationals. When variables range over the integers, treating the domain as $\mathbb{Q}$ is safe, but completeness is not guaranteed. Consider the following loop:

$$\texttt{while } (x_2 - x_1 \leq 0,\ x_1 + x_2 \geq 1) \texttt{ do } x_2' = x_2 - 2x_1 + 1,\ x_1' = x_1 \tag{2}$$

and observe that it does not terminate over the rationals at all (try $x_1 = x_2 = \frac{1}{2}$); but it has a *LRF* that is valid for all integer valuations, e.g., $f(x_1, x_2) = x_1 + x_2$. Several authors noted this issue, and finally the complexity of a complete solution for the integers was settled by [4], who proved that the detection problem is coNP-*complete* and gave matching algorithms.

However, not all terminating loops have a *LRF*; and to handle more loops, one may resort to an argument that combines several *LRFs* to capture a more complex behavior. Two types of such behavior that re-occur in the literature on termination are lexicographic ranking and multiphase ranking.

Lexicographic Ranking. One can prove the termination of a loop by considering a tuple, say a pair $\langle f_1, f_2 \rangle$ of linear functions, such that either f_1 decreases, or f_1 does not change and f_2 decreases. There are some variants of the definition [2,4,5,17] regarding whether both functions have to be non-negative at all times, or "just when necessary." The most permissive definition [17] allows any component to be negative, and technically, it ranks states in the lexicographic extension of the order $\succeq$ mentioned above. We refer to this class as *LLRFs*. For example, the following loop

$$\texttt{while } (x \geq 0, y \leq 10, z \geq 0, z \leq 1) \texttt{ do } x' = x+y+z-10, y' = y+z, z' = 1-z \tag{3}$$

has the *LLRF* $\langle 4y, 4x - 4z + 1 \rangle$, which is valid only according to [17].

Multiphase Ranking. Consider loop (1) above. Clearly, the loop goes through three phases—in the first, z descends, while the other variables may increase; in the second (which begins once z becomes negative), y decreases; in the last phase (beginning when y becomes negative), x decreases. Note that since there is no lower bound on y or on z, they cannot be used in a *LRF*; however, each phase is clearly finite, as it is associated with a value that is non-negative and decreasing during that phase. In other words, each phase is linearly ranked. We shall say that this loop has the *multiphase ranking function* ($M\Phi$RF) $\langle z+1, y+1, x \rangle$. The general definition (Sect. 2) allows for an arbitrary number d of linear components; we refer to d as *depth*, intuitively it is the number of phases.

Some loops have multiphase behavior which is not so evident as in the last example. Consider the following loop, that we will discuss further in Sect. 6, with $M\Phi$RF $\langle x - 4y, x - 2y, x - y \rangle$

$$\texttt{while } (x \geq 1,\ y \geq 1,\ x \geq y,\ 4y \geq x) \texttt{ do } x' = 2x,\ y' = 3y \tag{4}$$

Technically, under which ordering is a $M\Phi$RF a ranking function? It is quite easy to see that the pairs used in the examples above descend in the lexicographic extension of $\succeq$. This means that $M\Phi$RFs are a sub-class of *LLRFs*. Note that, intuitively, a lexicographic ranking function also has "phases", namely, steps where the first component decreases, steps where the second component decreases, etc.; but these phases may alternate an unbounded number of times.

Complete Solutions and Complexity. Complete solutions for $M\Phi$RFs (over the rationals) appear in [18,20]. Both use non-linear constraint solving, and therefore do not achieve a polynomial time complexity. [3] study "eventual linear ranking functions," which are $M\Phi$RFs of depth 2, and pose the questions of a polynomial-time solution and complete solution for the integers as open problems.

In this paper, we provide complete solutions to the existence and synthesis problems for both $M\Phi$RFs and *LLRFs*, for rational and integer *SLC* loops, where the algorithm is parameterized by a depth bound. Over the rationals, the decision problem is PTIME and the synthesis can be done in polynomial time; over the integers, the existence problem is `coNP`-complete, and our synthesis procedure is deterministic exponential-time.

While such algorithms would be a contribution in itself, we find it even more interesting that our results are mostly based on discovering unexpected *equivalences* between classes of ranking functions. We prove two such results: Theorem 3 in Sect. 4 shows that *LLRFs* are not stronger than $M\Phi$RFs for *SLC* loops. Thus, the complete solution for *LLRFs* is just to solve for $M\Phi$RFs (for the loop (3), we find the $M\Phi$RF $\langle 4y + x - z, 4x - 4z + 4\rangle$). Theorem 1 in Sect. 3 shows that one can further reduce the search for $M\Phi$RFs to a proper sub-class, called *nested* $M\Phi$RFs. This class was introduced in [18] because its definition is simpler and allows for a polynomial-time solution (over $\mathbb{Q}$)[2].

Our complete solution for the *integers* is also a reduction—transforming the problem so that solving over the rationals cannot give false alarms. The transformation consists of computing the *integer hull* of the transition polyhedron. This transformation is well-known in the case of *LRFs* [4,12,14], so it was a natural approach to try, however its proof in the case of $M\Phi$RFs is more involved.

We also make a contribution towards the use of $M\Phi$RFs in deriving *iteration bounds.* As the loop (1) demonstrates, it is possible for the variables that control subsequent phases to grow (at a polynomial rate) during the first phase. Nonetheless, we prove that *any* $M\Phi$RF *implies a linear bound on the number of iterations* for a *SLC* loop (in terms of the initial values of the variables). Thus, it is also the case that any *LLRF* implies a linear bound.

An open problem raised by our work is whether one can precompute a bound on the depth of a $M\Phi$RF for a given loop (if there is one); for example [4] prove a depth bound of n (the number of variables) on their notion of *LLRFs* (which is more restrictive); however their class is known to be weaker than $M\Phi$RFs and *LLRFs*. In Sect. 6 we discuss this problem.

The article is organized as follows. Section 2 gives precise definitions and necessary background. Sections 3 and 4 give our equivalence results for different types of ranking functions (over the rationals) and the algorithmic implications.

---

[2] This definition is also implicit in [6], where multi-path loops are considered, but each path should have a nested $M\Phi$RF.

Section 5 covers the integer setting, Sect. 6 discusses depth bounds, Sect. 7 discusses the iteration bound, and Sect. 8 concludes. For closely related work, see the above-mentioned references, while for further background on algorithmic and complexity aspects of linear/lexicographic ranking, we refer the reader to [4].

2 Preliminaries

In this section we define the class of loops we study, the type of ranking functions, and recall some definitions, and properties, regarding (integer) polyhedra.

Single-Path Linear-Constraint Loops. A *single-path* linear-constraint loop (*SLC* for short) over n variables $x_1, \ldots, x_n$ has the form

$$\texttt{while } (B\mathbf{x} \leq \mathbf{b}) \texttt{ do } A\binom{\mathbf{x}}{\mathbf{x}'} \leq \mathbf{c}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ and $\mathbf{x}' = (x'_1, \ldots, x'_n)^{\mathrm{T}}$ are column vectors, and for some $p, q > 0$, $B \in \mathbb{Q}^{p \times n}$, $A \in \mathbb{Q}^{q \times 2n}$, $\mathbf{b} \in \mathbb{Q}^p$, $\mathbf{c} \in \mathbb{Q}^q$. The constraint $B\mathbf{x} \leq \mathbf{b}$ is called *the loop condition* (a.k.a. the loop guard) and the other constraint is called *the update*. We say that the loop is a *rational loop* if $\mathbf{x}$ and $\mathbf{x}'$ range over $\mathbb{Q}^n$, and that it is an *integer loop* if they range over $\mathbb{Z}^n$. One could also allow variables to take any real-number value, but as long as the constraints are expressed by rational numbers, our results for $\mathbb{Q}$ also apply to $\mathbb{R}$.

We say that there is a transition from a state $\mathbf{x} \in \mathbb{Q}^n$ to a state $\mathbf{x}' \in \mathbb{Q}^n$, if $\mathbf{x}$ satisfies the condition and $\mathbf{x}$ and $\mathbf{x}'$ satisfy the update. A transition can be seen as a point $\binom{\mathbf{x}}{\mathbf{x}'} \in \mathbb{Q}^{2n}$, where its first n components correspond to $\mathbf{x}$ and its last n components to $\mathbf{x}'$. For ease of notation, we denote $\binom{\mathbf{x}}{\mathbf{x}'}$ by $\mathbf{x}''$. The set of all transitions $\mathbf{x}'' \in \mathbb{Q}^{2n}$, of a given *SLC* loop, will be denoted by $\mathcal{Q}$ and is specified by the constraints in the loop body and update. It is a polyhedron (see below), which we call *the transition polyhedron*. For the purpose of this article, the essence of the loop is this polyhedron, even if the loop is presented in a more readable form as above.

Multi-phase Ranking Functions. An affine function $f : \mathbb{Q}^n \to \mathbb{Q}$ is of the form $f(\mathbf{x}) = \vec{a} \cdot \mathbf{x} + a_0$ where $\vec{a} \in \mathbb{Q}^n$ is a row vector and $a_0 \in \mathbb{Q}$. For a given function f, we define the function $\Delta f : \mathbb{Q}^{2n} \mapsto \mathbb{Q}$ as $\Delta f(\mathbf{x}'') = f(\mathbf{x}) - f(\mathbf{x}')$.

Definition 1. *Given a set of transitions $T \subseteq \mathbb{Q}^{2n}$, we say that $\tau = \langle f_1, \ldots, f_d \rangle$ is a MΦRF (of depth d) for T if for every $\mathbf{x}'' \in T$ there is an index $i \in [1, d]$ such that:*

$$\forall j \leq i.\ \Delta f_j(\mathbf{x}'') \geq 1, \tag{5}$$
$$f_i(\mathbf{x}) \geq 0, \tag{6}$$
$$\forall j < i. \quad f_j(\mathbf{x}) \leq 0. \tag{7}$$

We say that $\mathbf{x}''$ is ranked by f_i *(for the minimal such i).*

It is not hard to see that this definition, for $d = 1$, means that f_1 is a linear ranking function, and for $d > 1$, it implies that as long as $f_1(\mathbf{x}) \geq 0$, transition $\mathbf{x}''$ must be ranked by f_1, and when $f_1(\mathbf{x}) < 0$, $\langle f_2, \ldots, f_d \rangle$ becomes a $M\Phi$RF. This agrees with the intuitive notion of "phases." We further note that, for loops specified by polyhedra, making the inequality (7) strict results in the same class of ranking functions (we chose the definition that is easier to work with), and, similarly, we can replace (5) by $\Delta f_j(\mathbf{x}'') > 0$, obtaining an equivalent definition (up to multiplication of the f_i by some constants). We say that τ is *irredundant* if removing any component invalidates the $M\Phi$RF.

The decision problem *Existence of a MΦRF* asks to determine whether a given *SLC* loop admits a $M\Phi$RF. The *bounded* decision problem, denoted by BMΦRF($\mathbb{Q}$) and BMΦRF($\mathbb{Z}$), restricts the search to $M\Phi$RFs of depth at most d, where the parameter d is part of the input.

Polyhedra. A *rational convex polyhedron* $\mathcal{P} \subseteq \mathbb{Q}^n$ (*polyhedron* for short) is the set of solutions of a set of inequalities $A\mathbf{x} \leq \mathbf{b}$, namely $\mathcal{P} = \{\mathbf{x} \in \mathbb{Q}^n \mid A\mathbf{x} \leq \mathbf{b}\}$, where $A \in \mathbb{Q}^{m\times n}$ is a rational matrix of n columns and m rows, $\mathbf{x} \in \mathbb{Q}^n$ and $\mathbf{b} \in \mathbb{Q}^m$ are column vectors of n and m rational values respectively. We say that $\mathcal{P}$ is specified by $A\mathbf{x} \leq \mathbf{b}$. If $\mathbf{b} = \mathbf{0}$, then $\mathcal{P}$ is a *cone*. The set of *recession directions* of a polyhedron $\mathcal{P}$ specified by $A\mathbf{x} \leq \mathbf{b}$, also know as its *recession cone*, is the set $\{\mathbf{y} \in \mathbb{Q}^n \mid A\mathbf{y} \leq \mathbf{0}\}$. For a given polyhedron $\mathcal{P} \subseteq \mathbb{Q}^n$ we let $I(\mathcal{P})$ be $\mathcal{P} \cap \mathbb{Z}^n$, i.e., the set of integer points of $\mathcal{P}$. The *integer hull* of $\mathcal{P}$, commonly denoted by $\mathcal{P}_I$, is defined as the convex hull of $I(\mathcal{P})$, i.e., every rational point of $\mathcal{P}_I$ is a convex combination of integer points. It is known that $\mathcal{P}_I$ is also a polyhedron, and that $\texttt{rec.cone}(\mathcal{P}) = \texttt{rec.cone}(\mathcal{P}_I)$ [23, Theorem 16.1, p. 231]. An *integer polyhedron* is a polyhedron $\mathcal{P}$ such that $\mathcal{P} = \mathcal{P}_I$. We also say that $\mathcal{P}$ is *integral*.

Next we state a lemma that is fundamental for many proofs in this article. Given a polyhedron $\mathcal{P}$, the lemma shows that if a disjunction of constraints of the form $f_i > 0$, or $f_i \geq 0$, holds over $\mathcal{P}$, then a certain conic combination of these functions is non-negative over $\mathcal{P}$.

Lemma 1. *Fix $\rhd$ to be either $>$ or $\geq$. Given a polyhedron $\mathcal{P} \neq \emptyset$, and linear functions $f_1, \ldots, f_k$ such that*

(*i*) $\mathbf{x} \in \mathcal{P} \rightarrow f_1(\mathbf{x}) \rhd 0 \vee \cdots \vee f_{k-1}(\mathbf{x}) \rhd 0 \vee f_k(\mathbf{x}) \geq 0$
(*ii*) $\mathbf{x} \in \mathcal{P} \not\rightarrow f_1(\mathbf{x}) \rhd 0 \vee \cdots \vee f_{k-1}(\mathbf{x}) \rhd 0$

There exist non-negative constants $\mu_1, \ldots, \mu_{k-1}$ such that $\mathbf{x} \in \mathcal{P} \rightarrow \mu_1 f_1(\mathbf{x}) + \cdots + \mu_{k-1} f_{k-1}(\mathbf{x}) + f_k(\mathbf{x}) \geq 0$.

Proof. We prove the lemma in one version; the other is very similar. Specifically, we assume:

(*i*) $\mathbf{x} \in \mathcal{P} \rightarrow f_1(\mathbf{x}) > 0 \vee \cdots \vee f_{k-1}(\mathbf{x}) > 0 \vee f_k(\mathbf{x}) \geq 0$
(*ii*) $\mathbf{x} \in \mathcal{P} \not\rightarrow f_1(\mathbf{x}) > 0 \vee \cdots \vee f_{k-1}(\mathbf{x}) > 0$.

Let $\mathcal{P}$ be $B\mathbf{x} \le \mathbf{c}$, $f_i = \vec{a}_i \cdot \mathbf{x} - b_i$, then (i) is equivalent to infeasibility of

$$B\mathbf{x} \le \mathbf{c} \wedge A\mathbf{x} \le \mathbf{b} \wedge \vec{a}_k \cdot \mathbf{x} < b_k \tag{8}$$

where A consists of the $k-1$ rows $\vec{a}_i$, and $\mathbf{b}$ of corresponding b_i. However, $B\mathbf{x} \le \mathbf{c} \wedge A\mathbf{x} \le \mathbf{b}$ is assumed to be feasible.

According to Motzkin's transposition theorem [23, Corollary 7.1k, p. 94], this implies that there are row vectors $\vec{\lambda}, \vec{\lambda}' \ge 0$ and a constant $\mu \ge 0$ such that the following is true:

$$\vec{\lambda}B + \vec{\lambda}'A + \mu a_k = 0 \wedge \vec{\lambda}\mathbf{c} + \vec{\lambda}'\mathbf{b} + \mu b_k \le 0 \wedge (\mu \ne 0 \vee \vec{\lambda}\mathbf{c} + \vec{\lambda}'\mathbf{b} + \mu b_k < 0) \tag{9}$$

Now, if (9) is true, then for all $\mathbf{x} \in \mathcal{P}$,

$$\begin{aligned}(\sum_i \lambda'_i f_i(\mathbf{x})) + \mu f_k(\mathbf{x}) &= \vec{\lambda}'A\mathbf{x} - \vec{\lambda}'\mathbf{b} + \mu a_k\mathbf{x} - \mu b_k \\ &= -\vec{\lambda}B\mathbf{x} - \vec{\lambda}'\mathbf{b} - \mu b_k \ge -\vec{\lambda}\mathbf{c} - \vec{\lambda}'\mathbf{b} - \mu b_k \ge 0\end{aligned}$$

where if $\mu = 0$, the last inequality must be strict. However, if $\mu = 0$, then $\vec{\lambda}B + \vec{\lambda}'A = 0$, so by feasibility of $B\mathbf{x} \le \mathbf{c}$ and $A\mathbf{x} \le \mathbf{b}$, this implies $\vec{\lambda}\mathbf{c} + \vec{\lambda}'\mathbf{b} \ge 0$, a contradiction. Thus, $(\sum_i \lambda'_i f_i) + \mu f_k \ge 0$ on $\mathcal{P}$ and $\mu > 0$. Dividing by μ we obtain the conclusion of the lemma. □

3 Complexity of Synthesis of *MΦ*RFs over the Rationals

In this section we study the complexity of deciding if a given rational *SLC* loop has a *MΦ*RF of depth d, and show that this can be done in polynomial time. These results follow from an equivalence between *MΦ*RFs and a sub-class called *nested ranking functions* [18]. In the rest of this article we assume a given *SLC* loop specified by a transition polyhedron $\mathcal{Q}$. The complexity results are in terms of the bit-size of the a constraint representation of $\mathcal{Q}$ (see Sect. 2 of [4]).

Definition 2. *A d-tuple $\tau = \langle f_1, \ldots, f_d \rangle$ is a* nested ranking function *for $\mathcal{Q}$ if the following are satisfied for all $\mathbf{x}'' \in \mathcal{Q}$ (where $f_0 \equiv 0$ for uniformity)*

$$f_d(\mathbf{x}) \ge 0 \tag{10}$$

$$(\Delta f_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}) \ge 0 \qquad \textit{for all } i = 1, \ldots, d. \tag{11}$$

It is easy to see that a nested ranking function is a *MΦ*RF. Indeed, f_1 is decreasing, and when it becomes negative f_2 starts to decrease, etc. Moreover, the loop must stop by the time that f_d becomes negative, since f_d is non-negative over $\mathcal{Q}$ (more precisely, on the projection of $\mathcal{Q}$ to its first n coordinates—as f_i is a function of state).

Example 1. Consider loop (1) (at Page 2). It has the *MΦ*RF $\langle z+1, y+1, x \rangle$ which is not nested because, among other things, last component x might be negative, e.g., for the state $x = -1, y = 0, z = 1$. However, it has the nested ranking function $\langle z+1, y+1, z+x \rangle$, which is *MΦ*RF.

The above example shows that there are $M\Phi$RFs which are not nested ranking functions, however, next we show that if a loop has a $M\Phi$RF then it has a (possibly different) nested ranking function of the same depth. We first state an auxiliary lemma, and then prove the main result.

Lemma 2. *Let* $\tau = \langle f_1, \ldots, f_d \rangle$ *be an irredundant* $M\Phi RF$ *for* $\mathcal{Q}$*, such that* $\langle f_2, \ldots, f_d \rangle$ *is a nested ranking function for* $\mathcal{Q}' = \mathcal{Q} \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid f_1(\mathbf{x}) \leq 0\}$*. Then there is a nested ranking function of depth* d *for* $\mathcal{Q}$*.*

Proof. First recall that, by definition of $M\Phi$RF, we have $\Delta f_1(\mathbf{x}'') \geq 1$ for any $\mathbf{x}'' \in \mathcal{Q}$, and since $\langle f_2, \ldots, f_d \rangle$ is a nested ranking function for $\mathcal{Q}'$ we have

$$\begin{aligned} \mathbf{x}'' \in \mathcal{Q}' \rightarrow & f_d(\mathbf{x}) \geq 0 \\ \mathbf{x}'' \in \mathcal{Q}' \rightarrow & (\Delta f_2(\mathbf{x}'') - 1) \geq 0 \wedge \\ & (\Delta f_3(\mathbf{x}'') - 1) + f_2(\mathbf{x}'') \geq 0 \wedge \\ & \cdots \\ & (\Delta f_d(\mathbf{x}'') - 1) + f_{d-1}(\mathbf{x}'') \geq 0 \end{aligned} \tag{12}$$

Next we construct a nested ranking function $\langle f'_1, \ldots, f'_d \rangle$ for $\mathcal{Q}$, i.e., such that (10) is satisfied for f'_d, and (11) is satisfied for each f'_i and f'_{i-1} — we refer to the instance of (11) for a specific i as (11_i).

We start with the condition (10). If f_d is non-negative over $\mathcal{Q}$ we let $f'_d = f_d$, otherwise, clearly $\mathbf{x}'' \in \mathcal{Q} \rightarrow f_d(\mathbf{x}) \geq 0 \vee f_1(\mathbf{x}) > 0$. Then, by Lemma 1 there is a constant $\mu_d > 0$ such that $\mathbf{x}'' \in \mathcal{Q} \rightarrow f_d(\mathbf{x}) + \mu_d f_1(\mathbf{x}) \geq 0$ and we define $f'_d(\mathbf{x}) = f_d(\mathbf{x}) + \mu_d f_1(\mathbf{x})$. Clearly (10) holds for f'_d.

Next, we handle the conditions (11_i) for $i = d, \ldots, 3$ in this order. When we handle (11_i), we shall define $f'_{i-1}(\mathbf{x}) = f_{i-1}(\mathbf{x}) + \mu_{i-1} f_1(\mathbf{x})$ for some $\mu_{i-1} \geq 0$. Note that f'_d has this form. Suppose we have computed $f'_d, \ldots, f'_i$. The construction of f'_{i-1} will ensure that (11_i) holds over $\mathcal{Q}$. From (12) we know that

$$\mathbf{x}'' \in \mathcal{Q}' \rightarrow (\Delta f_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}'') \geq 0.$$

Now since $f'_i(\mathbf{x}) = f_i(\mathbf{x}) + \mu_i f_1(\mathbf{x})$, and $\Delta f_1(\mathbf{x}'') \geq 1$ over $\mathcal{Q}$, we have

$$\mathbf{x}'' \in \mathcal{Q}' \rightarrow (\Delta f'_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}'') \geq 0.$$

Now if $(\Delta f'_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}'') \geq 0$ holds over $\mathcal{Q}$ as well, we let $f'_{i-1} = f_{i-1}$. Otherwise, we have

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow (\Delta f'_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}) \geq 0 \vee f_1(\mathbf{x}) > 0,$$

and by Lemma 1 there is $\mu_{i-1} > 0$ such that

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow (\Delta f'_i(\mathbf{x}'') - 1) + f_{i-1}(\mathbf{x}) + \mu_{i-1} f_1(\mathbf{x}) \geq 0.$$

In this case, we let $f'_{i-1}(\mathbf{x}) = f_{i-1}(\mathbf{x}) + \mu_{i-1} f_1(\mathbf{x})$. Clearly ($11_i$) holds.

We proceed to (11_2). From (12) we know that

$$\mathbf{x}'' \in \mathcal{Q}' \rightarrow (\Delta f_2(\mathbf{x}'') - 1) \geq 0.$$

Since $f_2'(\mathbf{x}) = f_2(\mathbf{x}) + \mu_2 f_1(\mathbf{x})$ and $\Delta f_1(\mathbf{x}'') \geq 1$ we have

$$\mathbf{x}'' \in \mathcal{Q}' \rightarrow (\Delta f_2'(\mathbf{x}'') - 1) \geq 0.$$

Next, by definition of $\mathcal{Q}'$ and the Lemma's assumption we have

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow (\Delta f_2'(\mathbf{x}'') - 1) \geq 0 \vee f_1(\mathbf{x}) > 0$$

and we also know that $(\Delta f_2'(\mathbf{x}'') - 1) \geq 0$ does not hold over $\mathcal{Q}$, because then f_1 would be redundant. Now by Lemma 1 there is $\mu_1 > 0$ such that

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow (\Delta f_2'(\mathbf{x}'') - 1) + \mu_1 f_1(\mathbf{x}) \geq 0.$$

We let $f_1'(\mathbf{x}) = \mu_1 f_1(\mathbf{x})$. For ($11_1$) we need to show that $\Delta f_1'(\mathbf{x}'') - 1 \geq 0$ holds over $\mathcal{Q}$, which clearly holds if $\mu_1 \geq 1$ since $\Delta f_1(\mathbf{x}'') \geq 1$; otherwise we multiply all f_i' by $\frac{1}{\mu_1}$, which does not affect any (11_i) and makes (11_1) true. □

Theorem 1. *If $\mathcal{Q}$ has a MΦRF of depth d, then it has a nested ranking function of depth at most d.*

Proof. The proof is by induction on d. We assume a *MΦ*RF $\langle f_1, \ldots, f_d \rangle$ for $\mathcal{Q}$. For $d = 1$ there is no difference between a general *MΦ*RF and a nested one. For $d > 1$, we consider $\langle f_2, \ldots, f_d \rangle$ as a *MΦ*RF for $\mathcal{Q}' = \mathcal{Q} \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid f_1(\mathbf{x}) \leq 0\}$, we apply the induction hypothesis to turn $\langle f_2, \ldots, f_d \rangle$ into a nested *MΦ*RF. Either f_1 becomes redundant, or we can apply Lemma 2. □

The above theorem gives us a complete algorithm for the synthesis of *MΦ*RFs of a given depth d for $\mathcal{Q}$, namely, just synthesize a nested *MΦ*RF.

Theorem 2. *BMΦRF*$(\mathbb{Q}) \in$ `PTIME`.

Proof. We describe, concisely, how to determine if a nested *MΦ*RF exists, and then synthesize one, in polynomial time (this actually appears in [18]). Given $\mathcal{Q}$, our goal is to find $f_1, \ldots, f_d$ such that (10), (11) hold. If we take just one of the conjuncts, our task is to find coefficients for the functions involved (f_d, or f_i and f_{i-1}), such that the desired inequality is implied by $\mathcal{Q}$. Using Farkas' lemma [23], this problem can be formulated as a *LP* problem, where the coefficients we seek are unknowns. By conjoing all these *LP* problems, we obtain a single *LP* problem, of polynomial size, whose solution—if there is one—provides the coefficients of all the f_i; and if there is no solution, then no nested *MΦ*RF exists. Since *LP* is polynomial-time, this procedure has polynomial time complexity. □

Clearly, if d is considered as constant, then BM*ΦRF*$(\mathbb{Q})$ is polynomial in the bit-size of $\mathcal{Q}$. When considering d as variable, then the complexity is polynomial in the bit-size of $\mathcal{Q}$ plus d—equivalently, it is polynomial in the bit-size of the input if we assume that d is given in unary representation (which is a reasonable assumption since d describes the number of components of the *MΦ*RF sought).

4 Multiphase vs Lexicographic-Linear Ranking Functions

*MΦ*RFs are similar to *LLRFs*, and a natural question is: which one is more powerful for proving termination of *SLC* loops? In this section we show that they have the same power, by proving that an *SLC* has a *MΦ*RF if and only if it has a *LLRF*. We first note that there are several definitions for *LLRFs* [2,4,5,17]. The following is the most general [17].

Definition 3. *Given a set of transitions* $T \subseteq \mathbb{Q}^{2n}$*, we say that* $\langle f_1, \ldots, f_d \rangle$ *is a LLRF (of depth d) for T if for every* $\mathbf{x}'' \in T$ *there is an index i such that:*

$$\forall j < i.\ \Delta f_j(\mathbf{x}'') \geq 0, \tag{13}$$

$$\Delta f_i(\mathbf{x}'') \geq 1, \tag{14}$$

$$f_i(\mathbf{x}) \geq 0, \tag{15}$$

We say that $\mathbf{x}''$ *is* ranked by f_i *(for the minimal such i).*

Regarding other definitions: [4] requires $f_j(\mathbf{x}) \geq 0$ for all f_j with $j \leq i$, and [2] requires it for all components. Actually [2] shows that an *SLC* loop has a *LLRF* according to their definition if and only if it has a *LRF*, which is not the case of [4]. The definition in [5] (which do not present here) is equivalent to a *LRF* when considering *SLC* loops, as their main interest is in multipath loops.

It is easy to see that a *MΦ*RF is also a *LLRF* as in Definition 3. Next we show that for *SLC* loops any *LLRF* can be converted to a *MΦ*RF, proving that these classes of ranking functions have the same power for *SLC* loops. We start with an auxiliary lemma.

Lemma 3. *Let f be a non-negative linear function over* $\mathcal{Q}$*. If* $\mathcal{Q}' = \mathcal{Q} \cap \{\mathbf{x}'' \mid \Delta f(\mathbf{x}'') \leq 0\}$ *has a MΦRF of depth d, then* $\mathcal{Q}$ *has one of depth at most* $d+1$*.*

Proof. Note that appending f to a *MΦ*RF τ of $\mathcal{Q}'$ does not always produce a *MΦ*RF, since the components of τ are not guaranteed to decrease over $\mathcal{Q} \setminus \mathcal{Q}'$. Let $\tau = \langle g_1, \ldots, g_d \rangle$ be a *MΦ*RF for $\mathcal{Q}'$, we construct a *MΦ*RF $\langle g'_1, \ldots, g'_d, f \rangle$ for $\mathcal{Q}$. If g_1 is decreasing over $\mathcal{Q}$, we define $g'_1(\mathbf{x}) = g_1(\mathbf{x})$. Otherwise, we have

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow \Delta f(\mathbf{x}'') > 0 \vee \Delta g_1(\mathbf{x}'') \geq 1.$$

Therefore, by Lemma 1, we can construct $g'_1(\mathbf{x}) = g_1(\mathbf{x}) + \mu f(\mathbf{x})$ such that $\mathbf{x}'' \in \mathcal{Q} \rightarrow \Delta g'_1(\mathbf{x}'') \geq 1$. Moreover, since f is non-negative, g'_1 is non-negative on the transitions on which g_1 is non-negative. If $d > 1$, we proceed with $\mathcal{Q}^{(1)} = \mathcal{Q} \cap \{\mathbf{x}'' \mid g'_1(\mathbf{x}) \leq (-1)\}$. Note that these transitions must be ranked, in $\mathcal{Q}'$, by $\langle g_2, \ldots, g_d \rangle$. If g_2 is decreasing over $\mathcal{Q}^{(1)}$, let $g'_2 = g_2$, otherwise

$$\mathbf{x}'' \in \mathcal{Q}^{(1)} \rightarrow \Delta f(\mathbf{x}'') > 0 \vee \Delta g_2(\mathbf{x}'') \geq 1,$$

and again by Lemma 1 we can construct the desired g'_2. In general for any $j \leq d$ we construct g'_{j+1} such that $\Delta g'_{j+1} \geq 1$ over

$$\mathcal{Q}^{(j)} = \mathcal{Q} \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid g'_1(\mathbf{x}) \leq (-1) \wedge \cdots \wedge g'_j(\mathbf{x}) \leq (-1)\}$$

and $\mathbf{x}'' \in \mathcal{Q} \wedge g_j(\mathbf{x}) \geq 0 \rightarrow g'_j(\mathbf{x}) \geq 0$. Finally we define

$$\mathcal{Q}^{(d)} = \mathcal{Q} \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid g'_1(\mathbf{x}) \leq (-1) \wedge g'_d(\mathbf{x}) \leq (-1)\},$$

each transition $\mathbf{x}'' \in \mathcal{Q}^{(d)}$ must satisfy $\Delta f(\mathbf{x}'') > 0$, and in such case $\Delta f(\mathbf{x}'')$ must have a minimum $c > 0$ since $\mathcal{Q}^{(d)}$ is a polyhedron. Without loss of generality we assume $c \geq 1$, otherwise take $\frac{1}{c}f$ instead of f. Now $\tau' = \langle g'_1+1, \ldots, g'_d+1, f\rangle$ is a $M\Phi$RF for $\mathcal{Q}$. Note that if we arrive to $\mathcal{Q}^{(j)}$ that is empty, we can stop since we already have a $M\Phi$RF. □

In what follows, by a *weak LLRF* we mean the class of ranking functions obtained by changing condition (14) to $\Delta f_i(\mathbf{x}'') > 0$. Clearly it is more general than *LLRFs*, and as we will see next it suffices to guarantee termination, since we show how to convert them to $M\Phi$RFs. We prefer to use this class as it simplifies the proof for the integer case that we present in Sect. 5.

Lemma 4. *Let $\langle f_1, \ldots, f_d\rangle$ be a* weak *LLRF for $\mathcal{Q}$. There is a linear function g that is positive over $\mathcal{Q}$, and decreasing on (at least) the same transitions of f_i, for some $1 \leq i \leq d$.*

Proof. If any f_i is positive over $\mathcal{Q}$, we take $g = f_i$. Otherwise, we have $\mathbf{x}'' \in \mathcal{Q} \rightarrow f_1(\mathbf{x}) \geq 0 \vee \cdots \vee f_d(\mathbf{x}) \geq 0$ since every transition is ranked by some f_i. If this implication satisfies the conditions of Lemma 1 then we can construct $g(\mathbf{x}) = f_d(\mathbf{x}) + \sum_{i=1}^{d-1} \mu_i f_i(\mathbf{x})$ that is non-negative over $\mathcal{Q}$, and, moreover, decreases on the transitions ranked by f_d. If the conditions of Lemma 1 are not satisfied, then the second condition must be false, that is, $\mathbf{x}'' \in \mathcal{Q} \rightarrow f_1(\mathbf{x}) \geq 0 \vee \cdots \vee f_{d-1}(\mathbf{x}) \geq 0$. Now we repeat the same reasoning as above for this implication. Eventually we either construct g that corresponds for some f_i as above, or we arrive to $\mathbf{x}'' \in \mathcal{Q} \rightarrow f_1(\mathbf{x}) \geq 0$, and then take $g = f_1$. □

Theorem 3. *If $\mathcal{Q}$ has a weak LLRF of depth d, it has a $M\Phi$RF of depth d.*

Proof. Let $\langle f_1, \ldots, f_d\rangle$ be a weak *LLRF* for $\mathcal{Q}$. We construct a corresponding $M\Phi$RF. The proof is by induction on the depth d of the *LLRF*. For $d = 1$ it is clear since it is an *LRF*.[3] Now let $d > 1$, by Lemma 4 we can find g that is positive over $\mathcal{Q}$ and decreasing at least on the same transitions as f_i. Now $\langle f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_d\rangle$ is a weak *LLRF* of depth $d-1$ for $\mathcal{Q}' = \mathcal{Q} \cap \{\mathbf{x}'' \mid \Delta g(\mathbf{x}'') \leq 0\}$. By the induction hypothesis we can construct a weak $M\Phi$RF for $\mathcal{Q}'$ of depth $d-1$, and by Lemma 3 we can lift it to one of depth d for $\mathcal{Q}$. □

Example 2. Let $\mathcal{Q}$ be the transition polyhedron of the loop (3) (on Page 3), which admits the *LLRF* $\langle 4y, 4x-4z+4\rangle$, and note that it is not a $M\Phi$RF. Following the proof of above theorem, we can convert it to the $M\Phi$RF $\langle 4y+x-z, 4x-4z+4\rangle$.

[3] If $\Delta f_i(\mathbf{x}'') > 0$ holds over $\mathcal{Q}$, then there must be $c > 0$ such that $\Delta f_i(\mathbf{x}'') \geq c$ holds over $\mathcal{Q}$. This is because a bounded LP problem (with non-strict inequalities only) attains its extremal value [23, p. 92].

5 *MΦ*RFs and *LLRFs* over the Integers

The procedure of Sect. 3 for synthesizing *MΦ*RFs, i.e., use *LP* to synthesize a nested ranking function, is complete for rational loops but not for integer loops. That is because it might be the case that $I(\mathcal{Q})$ has a *MΦ*RF but $\mathcal{Q}$ does not.

Example 3. Consider the loop

$$\texttt{while}\ (x_2 - x_1 \leq 0, x_1 + x_2 \geq 1, x_3 \geq 0)\ \texttt{do}\ x_2' = x_2 - 2x_1 + 1; x_3' = x_3 + 10x_2 + 9$$

When interpreted over the integers, it has the *MΦ*RF $\langle 10x_2, x_3 \rangle$. However, when interpreted over the rationals, the loop does not even terminate, e.g., for $(\frac{1}{2}, \frac{1}{2}, 0)$.

For *LRFs*, completeness for the integer case was achieved by reducing the problem to the rational case, using the integer hull $\mathcal{Q}_I$ [4,12]. In fact, it is quite easy to see why this reduction works for *LRFs*, as the requirements that a *LRF* has to satisfy are a conjunction of linear inequalities and if they are satisfied by $I(\mathcal{Q})$, they will be satisfied by convex combinations of such points, i.e., $\mathcal{Q}_I$.

Since we have reduced the problem of finding a *MΦ*RF to finding a nested ranking function, and the requirements from a nested ranking function are conjunctions of linear inequalities that should be implied by $\mathcal{Q}$, it is tempting to assume that this argument applies also for *MΦ*RFs. However, to justify the use of nested functions, specifically in proving Lemma 2, we relied on Lemma 1, which we applied to $\mathcal{Q}$ (it is quite easy to see that the lemma fails if instead of quantifying over a polyhedron, one quantifies only on its integer points). This means that we did not prove that the existence of a *MΦ*RF for $I(\mathcal{Q})$ implies the existence of a nested ranking function over $I(\mathcal{Q})$. A similar observation also holds for the results of Sect. 4, where we proved that any (weak) *LLRF* can be converted to a *MΦ*RF. Those results are valid only for rational loops, since in the corresponding proofs we used Lemma 1.

In this section we show that reduction of the integer case to the rational one, via the integer hull, does work also for *MΦ*RFs, and for converting *LLRFs* to *MΦ*RFs, thus extending our result to integer loops. We do so by showing that if $I(\mathcal{Q})$ has a weak *LLRF*, then $\mathcal{Q}_I$ has a weak *LLRF* (over the rationals).

Theorem 4. *Let $\langle f_1, \dots, f_d \rangle$ be a weak LLRF for $I(\mathcal{Q})$. There are constants $c_1, \dots, c_d$ such that $\langle f_1 + c_1, \dots, f_d + c_d \rangle$ is a weak LLRF for $\mathcal{Q}_I$.*

Proof. The proof is by induction on d. The base case, $d = 1$, concerns a *LRF*, and as already mentioned, is trivial (and $c_1 = 0$). For $d > 1$, define:

$$\boxed{\mathcal{Q}' = \mathcal{Q}_I \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid f_1(\mathbf{x}) \leq -1\}} \quad \boxed{\mathcal{Q}'' = \mathcal{Q}_I \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid \Delta f_1(\mathbf{x}'') = 0\}}$$

Note that $\mathcal{Q}'$ includes only integer points of $\mathcal{Q}_I$ that are not ranked at the first component, due to violating $f_1(\mathbf{x}) \geq 0$. By changing the first component into $f_1 + 1$, we take care of points where $-1 < f_1(\mathbf{x}) < 0$. Thus we will have that for every integer point $\mathbf{x}'' \in \mathcal{Q}$, if it is not in $\mathcal{Q}'$, then the first component is non-negative, and otherwise $\mathbf{x}''$ is ranked by $\langle f_2, \dots, f_d \rangle$. Similarly $\mathcal{Q}''$ includes all the integer points of $\mathcal{Q}_I$ that are not ranked by the first component due to violating $\Delta f_1(\mathbf{x}'') > 0$. Note also that $\mathcal{Q}''$ is integral, since it is a face of $\mathcal{Q}_I$. On the other hand, $\mathcal{Q}'$ is not necessarily integral, so we have to distinguish $\mathcal{Q}'_I$ from $\mathcal{Q}'$. By the induction hypothesis there are

- $c'_2, \ldots, c'_d$ such that $\langle f_2 + c'_2, \ldots, f_d + c'_d \rangle$ is a weak *LLRF* for $\mathcal{Q}'_I$; and
- $c''_2, \ldots, c''_d$ such that $\langle f_2 + c''_2, \ldots, f_d + c''_d \rangle$ is a weak *LLRF* for $\mathcal{Q}''_I$.

Next we prove that f_1 has a lower bound on $\mathcal{Q}_I \setminus \mathcal{Q}'_I$, i.e., there is $c_1 \geq 1$ such that $f_1 + c_1$ is non-negative on this set. Before proceeding to the proof, note that this implies that $\langle f_1 + c_1, f_2 + \max(c'_2, c''_2), \ldots, f_d + \max(c'_d, c''_d)\rangle$ is a weak *LLRF* for $\mathcal{Q}_I$. To see this, take any rational $\mathbf{x}'' \in \mathcal{Q}_I$, then either $\mathbf{x}''$ is ranked by the first component, or $\mathbf{x}'' \in \mathcal{Q}''$ or $\mathbf{x}'' \in \mathcal{Q}'_I$; in the last two cases, it is ranked by a component $f_i + \max(c'_i, c''_i)$ for $i > 1$.

It remains to prove that f_1 has a lower bound on $\mathcal{Q}_I \setminus \mathcal{Q}'_I$. We assume that $\mathcal{Q}'_I$ is non-empty, since otherwise, by the definition of $\mathcal{Q}'$, it is easy to see that $f_1 \geq -1$ over all of $\mathcal{Q}_I$. Thus, we consider $\mathcal{Q}'_I$ in an irredundant constraint representation: $\mathcal{Q}'_I = \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid \vec{a}_i \cdot \mathbf{x}'' \leq b_i,\ i = 1, \ldots, m\}$, and define

$$\boxed{\mathcal{P}_i = \mathcal{Q}_I \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid \vec{a}_i \cdot \mathbf{x}'' > b_i\}} \quad \boxed{\mathcal{P}'_i = \mathcal{Q}_I \cap \{\mathbf{x}'' \in \mathbb{Q}^{2n} \mid \vec{a}_i \cdot \mathbf{x}'' \geq b_i\}}$$

for $i = 1, \ldots, m$. Then, clearly, $\mathcal{Q}_I \setminus \mathcal{Q}'_I \subseteq \bigcup_{i=1}^m \mathcal{P}_i$. It suffices to prove that f_1 has a lower bound over $\mathcal{P}_i$, for every i. Fix i, such that $\mathcal{P}_i$ is not empty. It is important to note that, by construction, all integer points of $\mathcal{P}_i$ are in $\mathcal{Q}_I \setminus \mathcal{Q}'_I$.

Let H be the half-space $\{\mathbf{x}'' \mid \vec{a}_i \cdot \mathbf{x} \leq b_i\}$. We first claim that $\mathcal{P}_i = \mathcal{P}'_i \setminus H$ contains an integer point. Equivalently, there is an integer point of $\mathcal{Q}_I$ not contained in H. There has to be such a point, for otherwise, $\mathcal{Q}_I$, being integral, would be contained in H, and $\mathcal{P}_i$ would be empty. Let $\mathbf{x}''_0$ be such a point.

Next, assume (by way of contradiction) that f_1 is *not* lower bounded on $\mathcal{P}_i$. Express f_1 as $f_1(\mathbf{x}) = \vec{\lambda} \cdot \mathbf{x} + \lambda_0$, then $\vec{\lambda} \cdot \mathbf{x}$ is not lower bounded on $\mathcal{P}_i$ and thus not on $\mathcal{P}'_i$. This means that $\mathcal{P}'_i$ is not a polytope, and thus can be expressed as $\mathcal{O} + \mathcal{C}$, where $\mathcal{O}$ is a polytope and $\mathcal{C}$ is a cone [23, Corollary 7.1b, p. 88]. There must be a rational $\mathbf{y}'' \in \mathcal{C}$ such that $\vec{\lambda} \cdot \mathbf{y} < 0$, since otherwise f_1 would be bounded on $\mathcal{P}'_i$.

For $k \in \mathbb{Z}_+$, consider the point $\mathbf{x}''_0 + k\mathbf{y}''$. Clearly it is in P'_i. Since $\mathbf{y}'' \in \mathcal{C}$, we have $\vec{a}_i \cdot \mathbf{y}'' \geq 0$; Since $\mathbf{x}''_0 \in \mathcal{P}_i$, we have $\vec{a}_i \cdot \mathbf{x}''_0 > b_i$; adding up, we get $\vec{a}_i \cdot (\mathbf{x}''_0 + k\mathbf{y}'') > b_i$ for all k. We conclude that the set $S = \{\mathbf{x}''_0 + k\mathbf{y}'' \mid k \in \mathbb{Z}_+\}$ is contained in $\mathcal{P}_i$. Clearly, it includes an infinite number of integer points. Moreover f_1 obtains arbitrarily negative values on S (the larger k, the smaller the value), in particular on its integer points. Recall that these points are included $\mathcal{Q}_I \setminus \mathcal{Q}'_I$, thus f_1 is not lower bounded on the *integer points of* $\mathcal{Q}_I \setminus \mathcal{Q}'_I$, a contradiction to the way $\mathcal{Q}'_I$ was defined. □

Corollary 1. *If $I(\mathcal{Q})$ has a weak LLRF of depth d, then $\mathcal{Q}_I$ has a MΦRF, of depth at most d.*

Proof. By Theorem 4 we know that $\mathcal{Q}_I$ has a weak *LLRF* (of the same depth), which in turn can be converted to a *MΦ*RF by Theorem 3. □

Since $M\Phi$RFs are also weak $LLRF$, the above corollary provides a complete procedure for synthesizing $M\Phi$RFs over the integers, simply by seeking a nested ranking function for $\mathcal{Q}_I$.

Example 4. For the loop of Example 3, computing the integer hull results in the addition of $x_1 \geq 1$. Now seeking a $M\Phi$RF as in Sect. 3 we find, for example, $\langle 10x_2 + 10, x_3 \rangle$. Note that $\langle 10x_2, x_3 \rangle$, which a $M\Phi$RF for $I(\mathcal{Q})$, is not a $M\Phi$RF for Q_I according to Definition 1, e.g., for any $0 < \varepsilon < 1$ the transition $(1 + \varepsilon, -\varepsilon, 0, 1, -3\varepsilon - 1, -10\varepsilon + 9) \in Q_I$ is not ranked, since $10x_2 < 0$ and $x_3 - x_3' = 10\varepsilon - 9 < 1$.

The procedure described above has exponential-time complexity, because computing the integer hull requires exponential time. However, it is polynomial for the cases in which the integer hull can be computed in polynomial time [4, Sect. 4]. The next theorem shows that the exponential time complexity is unavoidable for the general case (unless $\mathtt{P} = \mathtt{NP}$). The proof repeats the arguments in the $\mathtt{coNP}$-completes proof for *LRFs* [4, Sect. 3]. We omit the details.

Theorem 5. *$BM\Phi RF(\mathbb{Z})$ is* $\mathtt{coNP}$*-complete.*

As in Sect. 3, we consider d as constant, or as input given in unary.

6 The Depth of a $M\Phi$RF

A wishful thought: If we could pre-compute an upper bound on the depth of optimal $M\Phi$RFs, and use it to bound the recursion, we would obtain a complete decision procedure for $M\Phi$RFs in general, since we can seek a $M\Phi$RF, as in Sect. 3, of this specific depth. This thought is motivated by results for *lexicographic ranking functions*, for example, [4] shows that the number of components in such functions is bounded by the number of variables in the loop. For $M\Phi$RFs, we were not able to find a similar upper bound, and we can show that the problem is more complicated than in the lexicographic case as a bound, if one exists, must depend not only on the number of variables or constraints, but also on the values of the coefficients in the loop constraints.

Theorem 6. *For integer $B > 0$, the following loop $\mathcal{Q}_B$*

$$\texttt{while } (x \geq 1,\ y \geq 1,\ x \geq y,\ 2^B y \geq x) \texttt{ do } x' = 2x,\ y' = 3y$$

needs at least $B+1$ components in any $M\Phi RF$.

Proof. Define $R_i = \{(2^i c, c, 2^{i+1}c, 3c) \mid c \geq 1\}$ and note that for $i = 0 \ldots B$, we have $R_i \subset \mathcal{Q}_B$. Moreover, $R_i \neq R_j$ for different i and j. Next we prove that in any $M\Phi$RF $\langle f_1, \ldots, f_d \rangle$ for $\mathcal{Q}_B$, and R_i with $i = 0 \ldots B$, there must be a component f_k such that $\mathbf{x}'' \in R_i \rightarrow f_k(\mathbf{x}) - f(0,0) \geq 0 \wedge \Delta f_k(\mathbf{x}'') > 0$. To prove this, fix i. We argue by the pigeonhole principle that, for some k, $f_k(2^i c, c) = c f_k(2^i, 1) + (1-c) f_k(0,0) \geq 0$ and $f_k(2^i c, c) - f_k(2^{i+1}c, 3c) = c(f_k(2^i, 1) - f_k(2^{i+1}, 3)) > 0$

for infinite number of values of c, and thus $f_k(2^i,1) - f_k(0,0) \geq 0$, and $f_k(2^i,1) - f_k(2^{i+1},3) > 0$, leading to the above statement. We say that R_i is "ranked" by f_k.

If $d < B+1$, then, by the pigeonhole principle, there are different R_i and R_j that are "ranked" by the same f_k. We show that this leads to a contradiction. Consider R_i and R_j, with $j > i$, and assume that they are "ranked" by $f_k(x,y) = a_1x + a_2y + a_0$. Applying the conclusion of the last paragraph to R_i and R_j, we have:

$$f_k(2^i,1) - f_k(2^{i+1},3) = -a_1 2^i - a_2 2 > 0 \quad (16)$$

$$f_k(2^j,1) - f_k(2^{j+1},3) = -a_1 2^j - a_2 2 > 0 \quad (17)$$

$$f_k(2^i,1) - f_k(0,0) = a_1 2^i + a_2 \geq 0 \quad (18)$$

$$f_k(2^j,1) - f_k(0,0) = a_1 2^j + a_2 \geq 0 \quad (19)$$

Adding $\frac{1}{2}\cdot$(17) to (19) we get $a_1 2^{j-1} > 0$. Thus, a_1 must be positive. From the sum of $\frac{1}{2}\cdot$(17) and (18), we get $a_1(2^i - 2^{j-1}) > 0$, which implies $j > i+1$, and $a_1 < 0$, a contradiction. Note that the bound $B+1$ is tight. This is confirmed by the *MΦ*RF $\langle x - 2^B y, x - 2^{B-1} y, x - 2^{B-2} y, \dots, x - y\rangle$. □

7 Iteration Bounds from *MΦ*RFs

Automatic complexity analysis techniques are often based on bounding the number of iterations of loops, using ranking functions. Thus, it is natural to ask if a *MΦ*RF implies a bound on the number of iterations of a given *SLC* loop. For *LRFs*, the implied bound is trivially linear, and in the case of *SLC* loops, it is known to be linear also for a class of lexicographic ranking functions [4]. In this section we show that *MΦ*RFs, too, imply a linear iteration bound, despite the fact that the variables involved may grow non-linearly during the loop. Below we concentrate on its existence, but the bound can also be computed explicitly.

Example 5. Consider the following loop with the corresponding *MΦ*RF $\langle y+1, x\rangle$

$$\texttt{while } (x \geq 0) \texttt{ do } x' = x + y,\ y' = y - 1$$

Let us consider an execution starting from positive values x_0 and y_0, and note that when $y+1$ reaches 0, i.e., when moving to the second phase, the value of x would be $x_0 - 1 + \sum_{i=-1}^{y_0} i = x_0 - 1 + \frac{y_0(y_0+1)}{2}$, which is polynomial in the input. It may seem that the next phase would be super-linear, since it is ranked by x, however, note that x decreases first by 1, then by 2, then by 3, etc. Therefore the quantity $\frac{y_0(y_0+1)}{2}$ is eliminated in y_0 iterations.

In what follows we generalize the observation of the above example. We consider an *SLC* loop $\mathcal{Q}$, and a corresponding irredundant *MΦ*RF $\tau = \langle f_1, \dots, f_d\rangle$. Let us start with an outline of the proof. We first define a function $F_k(t)$ that corresponds to the value of f_k after iteration t. We then bound each F_k by some expression $UB_k(t)$, and observe that for t greater than some number T_k, that depends linearly on the input, $UB_k(T_k)$ becomes negative. This means that T_k is an upper bound on the time in which the k-th phase ends; the whole loop must terminate before $\max_k T_k$ iterations.

Let $\mathbf{x}_t$ be the state after iteration t, and define $F_k(t) = f_k(\mathbf{x}_t)$, i.e., the value of the k-th component f_k after t iterations. For the initial state $\mathbf{x}_0$, we let $M = \max\{f_1(\mathbf{x}_0), \dots, f_d(\mathbf{x}_0), 1\}$. Note that M is linear in $\|\mathbf{x}_0\|_\infty$ (i.e., in the maximum absolute value of the components of $\mathbf{x}_0$). We first state an auxiliary lemma, and then a lemma that bounds F_k.

Lemma 5. *For all $1 < k \le d$, there are $\mu_1, \dots, \mu_{k-2} \ge 0$ and $\mu_{k-1} > 0$ such that $\mathbf{x}'' \in \mathcal{Q} \rightarrow \mu_1 f_1(\mathbf{x}) + \cdots + \mu_{k-1} f_{k-1}(\mathbf{x}) + (\Delta f_k(\mathbf{x}'') - 1) \ge 0$.*

Proof. From the definition of $M\Phi$RF we have

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow f_1(\mathbf{x}) \ge 0 \vee \cdots \vee f_{k-1}(\mathbf{x}) \ge 0 \vee \Delta f_k(\mathbf{x}'') \ge 1.$$

Moreover the conditions of Lemma 1 hold since f_k is not redundant, thus there are non-negative constants $\mu_1, \dots, \mu_{k-1}$ such that

$$\mathbf{x}'' \in \mathcal{Q} \rightarrow \mu_1 f_1(\mathbf{x}) + \cdots + \mu_{k-1} f_{k-1}(\mathbf{x}) + (\Delta f_k(\mathbf{x}'') - 1) \ge 0.$$

Moreover, at least μ_{k-1} must be non-zero, otherwise it means that $\Delta f_k(\mathbf{x}'') \ge 1$ holds already when $f_1, \dots, f_{k-2}$ are negative, so f_{k-1} would be redundant. □

Lemma 6. *For all $1 \le k \le d$, there are constants $c_k, d_k > 0$ such that $F_k(t) \le c_k M t^{k-1} - d_k t^k$, for all $t \ge 1$.*

Proof. The proof is by induction. For $k = 1$ we let $c_1 = d_1 = 1$ and get $F_1(t) \le M - t$, which is trivially true. For $k \ge 2$ we assume that the lemma holds for smaller indexes and show that it holds for k. Note that the change in the value of $F_k(t)$ in the i-th iteration is $f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_i) = -\Delta f_k(\mathbf{x}''_i)$. By Lemma 5 and the definition of F_k, we have $\mu_1, \dots, \mu_{k-2} \ge 0$ and $\mu_{k-1} > 0$ such that (over $\mathcal{Q}$)

$$f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_i) < \mu_1 F_1(i) + \cdots + \mu_{k-1} F_{k-1}(i)\,. \tag{20}$$

Now we bound F_k (explanation follows):

$$\begin{aligned}
F_k(t) &= f_k(\mathbf{x}_0) + \Sigma_{i=0}^{t-1}(f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_i)) && (21)\\
&< M + \Sigma_{i=0}^{t-1}\,(\mu_1 F_1(i) + \cdots + \mu_{k-1} F_{k-1}(i)) && (22)\\
&\le M(1+\mu) + \Sigma_{i=1}^{t-1}\,(\mu_1 F_1(i) + \cdots + \mu_{k-1} F_{k-1}(i)) && (23)\\
&\le M(1+\mu) + \Sigma_{i=1}^{t-1} \Sigma_{j=1}^{k-1} \left(\mu_j c_j M i^{j-1} - \mu_j d_j i^j\right) && (24)\\
&\le M(1+\mu) + \Sigma_{i=1}^{t-1} \left((\Sigma_{j=1}^{k-1} \mu_j c_j M i^{j-1}) - \mu_{k-1} d_{k-1} i^{k-1}\right) && (25)\\
&\le M(1+\mu) + \Sigma_{i=1}^{t-1} \left(M(\Sigma_{j=1}^{k-1} \mu_j c_j) i^{k-2} - \mu_{k-1} d_{k-1} i^{k-1}\right) && (26)\\
&= M(1+\mu) + M(\Sigma_{j=1}^{k-1} \mu_j c_j)(\Sigma_{i=1}^{t-1} i^{k-2}) - \mu_{k-1} d_{k-1} \Sigma_{i=1}^{t-1} i^{k-1} && (27)\\
&\le M(1+\mu) + M(\Sigma_{j=1}^{k-1} \mu_j c_j)(\frac{t^{k-1}}{k-1}) - \mu_{k-1} d_{k-1}(\frac{t^k}{k} - t^{k-1}) && (28)\\
&= c_k M t^{k-1} - d_k t^k && (29)
\end{aligned}$$

Each step above is obtained from the previous one as follows: (22) by replacing $f_k(\mathbf{x}_0)$ by M, since $f_k(\mathbf{x}_0) \le M$, and applying (20); (23) by separating the term for $i = 0$ from the sum; this term is bounded by μM, where $\mu = \sum_{j=1}^{k-1} \mu_j$, because $F_k(0) = f_k(\mathbf{x}_0) \le M$ by definition; (24) by applying the induction hypothesis; (25) by removing all negative values $-\mu_j d_j i^j$, except the last one $-\mu_{k-1} d_{k-1} i^{k-1}$; (26) by replacing i^{j-1} by an upper bound i^{k-2}; (27) by opening parentheses; (28) replacing $\Sigma_{i=1}^{t-1} i^{k-2}$ by an upper bound $\frac{t^{k-1}}{k-1}$, and $\Sigma_{i=1}^{t-1} i^{k-1}$ by a lower bound $\frac{t^k}{k} - t^{k-1}$; finally for (28), take $c_k = (1+\mu) + (\Sigma_{j=1}^{k-1} \mu_j c_j)/(k-1)$ $+ \mu_{k-1} d_{k-1}$, and $d_k = \mu_{k-1} d_{k-1}/k$ and note that both are positive. □

Theorem 7. *An SLC loop that has a MΦRF terminates in a number of iterations bounded by* $O(\|\mathbf{x}_0\|_\infty)$.

Proof. For $t > \max\{1, (c_k/d_k)M\}$, we have $F_k(t) < 0$, proving that the k-th phase terminates by this time (since it remains negative). Thus, by the time $\max\{1, (c_1/d_1)M, \ldots, (c_d/d_d)M\}$, which is linear in $\|\mathbf{x}_0\|_\infty$, all phases must have terminated. Note that c_k and d_k can be computed explicitly, if desired. □

We remark that the above result also holds for multi-path loops if they have a nested *MΦ*RF, but does not hold for any *MΦ*RF.

8 Conclusion

LRFs, *LLRFs* and *MΦ*RFs have all been proposed in earlier work. The original purpose of this work has been to improve our understanding of *MΦ*RFs, and answer open problems regarding the complexity of obtaining such functions from linear-constraint loops, the difference between the integer case and the rational case, and the possibility of inferring an iteration bound from such ranking functions. Similarly, we wanted to understand a natural class of lexicographic ranking functions, which removes a restriction of previous definitions regarding negative values. Surprisingly, it turned out that our main results are *equivalences* which show that, for *SLC* loops, both *MΦ*RFs and *LLRFs* reduce to a simple kind of *MΦ*RFs, that has been known to allow polynomial-time solution (over $\mathbb{Q}$). Thus, our result collapsed, in essence, these above classes of ranking functions.

The implication of having a polynomial-time solution, which is hardly more complex than the standard algorithm to find *LRFs*, is that whenever one considers using *LRFs* in one's work, one should consider using *MΦ*RFs. By controlling the depth of the *MΦ*RFs one trades expressivity for processing cost. We believe that it would be sensible to start with depth 1 (i.e., seeking a *LRF*) and increase the depth upon failure. Similarly, since a complete solution for the integers is inherently more costly, it makes sense to begin with the solution that is complete over the rationals, since it is safe for the integer case. If this fail, one can also consider special cases in which the inherent hardness can be avoided [4, Sect. 4].

Theoretically, some tantalizing open problems remain. Is it possible to decide whether a given loop admits a *MΦ*RF, without a depth bound? This is related to

Table 1. Experiments on loops taken from [15]—loops (1–5) originate from [11] and (6–41) originate from [8].

♯	loop	Nested $M\Phi$RF
1	`while (x≥0) x'=-2x+10;`	None
2	`while (x>0) x'=x+y; y'=y+z;`	None
3	`while (x≤N)` `  if (*) { x'=2*x+y; y'=y+1; } else x'=x+1;`	None
4	`while (1)` `  if (x<n) { x'=x+y; if (x'≥200) break; }`	None
5	`while (x<>y) if (x>y) x'=x-y; else y'=y-x;`	None
6	`while (x<0) x'=x+y; y'=y-1;`	None
7	`while (x>0) x'=x+y; y'=-2y;`	None
8	`while (x<y) x'=x+y; y'=-2y;`	None
9	`while (x<y) x'=x+y; 2y'=y;`	None
10	`while (4x-5y>0) x'=2x+4y; y'=4x;`	None
11	`while (x<5) x'=x-y; y'=x+y;`	None
12	`while (x>0, y>0) x'=-2x+10y;`	None
13	`while (x>0) x'=x+y;`	None
14	`while (x<10) x'=-y; y'=y+1;`	None
15	`while (x<0) x'=x+z; y'=y+1; z'=-2y`	None
16	`while (x>0, x<100) x'≥2x+10;`	$\langle -\frac{1}{11}x + \frac{111}{11} \rangle$
17	`while (x>1) -2x'=x;`	$\langle \frac{2}{4}x \rangle$
18	`while (x>1) 2x'≤x;`	$\langle x \rangle$
19	`while (x>0) 2x'≤x;`	$\langle 2x \rangle$
20	`while (x>0) x'=x+y; y'=y-1;`	$\langle y+1, x \rangle$
21	`while (4x+y>0) x'=-2x+4y; y'=4x;`	None
22	`while (x>0, x<y) x'=2x; y'=y+1;`	$\langle -3x+2y, -x+y \rangle$
23	`while (x>0) x'=x-2y; y'=y+1;`	$\langle -2y+1, x \rangle$
24	`while (x>0, x<n) x'=-x+y-5; y'=2y; n'=n;`	$\langle \frac{2}{14}x - \frac{1}{7}y + \frac{13}{14}n, -\frac{2}{14}x + \frac{8}{14}n \rangle$
25	`while (x>0, y<0) x'=x+y; y'=y-1;`	$\langle x \rangle$
26	`while (x-y>0) x'=-x+y; y'=y+1;`	$\langle -y, x-y \rangle$
27	`while (x>0) x'=y; y'=y-1;`	$\langle y, x \rangle$
28	`while (x>0) x'=x+y-5; y'=-2y;`	$\langle \frac{3}{8}x + \frac{1}{8}y, \frac{1}{8}x + \frac{7}{8} \rangle$
29	`while (x+y>0) x'=x-1; y'=-2y;`	$\langle x, \frac{1}{3}x + \frac{1}{3}y + \frac{2}{3} \rangle$
30	`while (x>y) x'=x-y; 1≤y'≤2`	$\langle x-y \rangle$
31	`while (x>0) x'=x+y; y'=-y-1;`	$\langle 2x+y, x \rangle$
32	`while (x>0) x'=y; y'≤-y;`	$\langle x + \frac{1}{2}y, \frac{1}{2}x + \frac{1}{2} \rangle$
33	`while (x<y) x'=x+1; y'=z; z'=z;`	$\langle -x+z, -x+y \rangle$
34	`while (x>0) x'=x+y; y'=y+z; z'=z-1;`	$\langle z+1, y+1, x \rangle$
35	`while (x+y≥0, x≤z) x'=2x+y; y'=y+1; z'=z`	$\langle -x-y+1, -x+z+1 \rangle$
36	`while (x>0, x≤z) x'=2x+y; y'=y+1; z'=z`	$\langle -y, -x+2z \rangle$
37	`while (x≥0) x'=x+y; y'=z; z'=-z-1;`	$\langle 2x+2y+z+1, 3x+y+1, x+1 \rangle$
38	`while (x-y>0) x'=-x+y; y'=z; z'=z+1;`	$\langle -z, x-y \rangle$
39	`while (x>0, x<y) x'>2x; y'=z; z'=z;`	$\langle -\frac{6}{12}x + \frac{7}{12}z, -\frac{2}{12}x + \frac{7}{12}y \rangle$
40	`while (x≥0, x+y≥0) x'=x+y+z; y'=-z-1; z'=z;`	$\langle x+y+1 \rangle$
41	`while (x+y≥0, x≤n) x'=2x+y; y'=z; z'=z+1; n'=n;`	$\langle -x-z+1, -x-y+1, -x+n+1 \rangle$

the question, discussed in Sect. 6, whether it is possible to precompute a depth bound. What is the complexity of the $M\Phi$RF problems over multi-path loops? For such loops, the equivalence of $M\Phi$RFs, nested r.f.s and *LLRFs* does not hold. Finally (generalizing the first question), we think that there is need for further exploration of single-path loops and of the plethora of "termination witnesses" based on linear functions (a notable reference is [18]).

We have implemented the *nested ranking function* procedure of Sect. 3, and applied it on a set of terminating and non-terminating *SLC* loops taken from [15]. Table 1 (at Page 18) summarizes these experiments. All loops marked with "`None`" have no $M\Phi$RF, and they are mostly non-terminating: Loop 1 is terminating over the integers, but does not have $M\Phi$RF, and Loop 21 is terminating over both integers and rationals but does not have $M\Phi$RF. Note that loops 3–5 are multi-path, in such case we seek a nested $M\Phi$RF that is valid for all paths. In all cases, a strict inequality $x > y$ was translated to $x \geq y + 1$ before applying out procedure. Interestingly, all 25 terminating loops in this set have a $M\Phi$RF. These experiments can be tried at http://loopkiller.com/irankfinder.

References

1. Albert, E., Arenas, P., Genaim, S., Puebla, G.: Closed-form upper bounds in static cost analysis. J. Autom. Reason. **46**(2), 161–203 (2011)
2. Alias, C., Darte, A., Feautrier, P., Gonnord, L.: Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs. In: Cousot, R., Martel, M. (eds.) SAS 2010. LNCS, vol. 6337, pp. 117–133. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15769-1_8
3. Bagnara, R., Mesnard, F.: Eventual linear ranking functions. In: Proceedings of the 15th International Symposium on Principles and Practice of Declarative Programming, PPDP 2013, pp. 229–238. ACM Press (2013)
4. Ben-Amram, A.M., Genaim, S.: Ranking functions for linear-constraint loops. J. ACM **61**(4), 26:1–26:55 (2014)
5. Bradley, A.R., Manna, Z., Sipma, H.B.: Linear ranking with reachability. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 491–504. Springer, Heidelberg (2005). doi:10.1007/11513988_48
6. Bradley, A.R., Manna, Z., Sipma, H.B.: The polyranking principle. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1349–1361. Springer, Heidelberg (2005). doi:10.1007/11523468_109
7. Brockschmidt, M., Emmes, F., Falke, S., Fuhs, C., Giesl, J.: Analyzing runtime and size complexity of integer programs. ACM Trans. Program. Lang. Syst. **38**(4), 13 (2016)
8. Chen, H.Y., Flur, S., Mukhopadhyay, S.: Termination proofs for linear simple loops. STTT **17**(1), 47–57 (2015)
9. Colóon, M.A., Sipma, H.B.: Synthesis of linear ranking functions. In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 67–81. Springer, Heidelberg (2001). doi:10.1007/3-540-45319-9_6
10. Cook, B., Gotsman, A., Podelski, A., Rybalchenko, A., Vardi, M.Y.: Proving that programs eventually do something good. In: Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, 17–19 January 2007, pp. 265–276 (2007)

11. Cook, B., Gulwani, S., Lev-Ami, T., Rybalchenko, A., Sagiv, M.: Proving conditional termination. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 328–340. Springer, Heidelberg (2008). doi:10.1007/978-3-540-70545-1_32
12. Cook, B., Kroening, D., Rümmer, P., Wintersteiger, C.M.: Ranking function synthesis for bit-vector relations. Formal Methods Syst. Des. **43**(1), 93–120 (2013)
13. Cook, B., Podelski, A., Rybalchenko, A.: Termination proofs for systems code. In: Schwartzbach, M.I., Ball, T., (eds.) Programming Language Design and Implementation, PLDI 2006, pp. 415–426. ACM (2006)
14. Feautrier, P.: Some efficient solutions to the affine scheduling problem I. One-dimensional time. Int. J. Parallel Prog. **21**(5), 313–347 (1992)
15. Ganty, P., Genaim, S.: Proving termination starting from the end. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 397–412. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39799-8_27
16. Harrison, M.: Lectures on Sequential Machines. Academic Press, Cambridge (1969)
17. Larraz, D., Oliveras, A., Rodríguez-Carbonell, E., Rubio, A.: Proving termination of imperative programs using Max-SMT. In: Formal Methods in Computer-Aided Design, FMCAD 2013, pp. 218–225. IEEE (2013)
18. Leike, J., Heizmann, M.: Ranking templates for linear loops. Log. Methods Comput. Sci. **11**(1), 1–27 (2015)
19. Leroux, J., Sutre, G.: Flat counter automata almost everywhere!. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 489–503. Springer, Heidelberg (2005). doi:10.1007/11562948_36
20. Li, Y., Zhu, G., Feng, Y.: The L-depth eventual linear ranking functions for single-path linear constraint loops. In: 10th International Symposium on Theoretical Aspects of Software Engineering (TASE 2016), pp. 30–37. IEEE (2016)
21. Ouaknine, J., Worrell, J.: On linear recurrence sequences and loop termination. ACM SIGLOG News **2**(2), 4–13 (2015)
22. Podelski, A., Rybalchenko, A.: A complete method for the synthesis of linear ranking functions. In: Steffen, B., Levi, G. (eds.) VMCAI 2004. LNCS, vol. 2937, pp. 239–251. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24622-0_20
23. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1986)
24. Sohn, K., Van Gelder, A.: Termination detection in logic programs using argument sizes. In: Rosenkrantz, D.J. (ed.) Symposium on Principles of Database Systems, pp. 216–226. ACM Press (1991)