Complexity of Bradley-Manna-Sipma Lexicographic Ranking Functions*

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

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

Abstract. In this paper we turn the spotlight on a class of lexicographic ranking functions introduced by Bradley, Manna and Sipma in a seminal CAV 2005 paper, and establish for the first time the complexity of some problems involving the inference of such functions for linear-constraint loops (without precondition). We show that finding such a function, if one exists, can be done in polynomial time in a way which is sound and complete when the variables range over the rationals (or reals). We show that when variables range over the integers, the problem is harder—deciding the existence of a ranking function is coNP-complete. Next, we study the problem of minimizing the number of components in the ranking function (a.k.a. the dimension). This number is interesting in contexts like computing iteration bounds and loop parallelization. Surprisingly, and unlike the situation for some other classes of lexicographic ranking functions, we find that even deciding whether a two-component ranking function exists is harder than the unrestricted problem: NP-complete over the rationals and Σ_2^P-complete over the integers.

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 ρ that maps the program states into the elements of a well-founded ordered set, such that $\rho(s) > \rho(s')$ holds for any consecutive states s and s'. This implies termination since infinite descent in a well-founded order is impossible.

We focus on *numerical loops*, where a state is described by the values of a finite set of numerical variables; we consider the setting of integer-valued variables, as well as rational-valued (or real-valued) variables. We ignore details of the programming language; we assume that we are provided an abstract description of the loop as a finite number of alternatives, that we call *paths*, each one defined by a finite set of *linear constraints* on the program variables $x, y, \ldots$ and

* This work was funded partially by the EU project FP7-ICT-610582 ENVISAGE: Engineering Virtualized Services (http://www.envisage-project.eu), by the Spanish MINECO project TIN2012-38137, and by the CM project S2013/ICE-3006.

the primed variables $x', y', \ldots$ which refer to the state following the iteration. The following is such a loop consisting of four paths, $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$:

$$\begin{array}{lllll}
\mathcal{Q}_1 = \{x \geq 0, & x' \leq x - 1, & y' = y, & & z' = z\} \\
\mathcal{Q}_2 = \{x \geq 0, & x' \leq x - 1, & y' = y, & z \geq 0, & z' \leq z - 1\} \\
\mathcal{Q}_3 = \{ & x' = x, & y \geq 0, y' \leq y - 1, & z \geq 0, & z' \leq z - 1\} \\
\mathcal{Q}_4 = \{ & x' = x, & y \geq 0, y' \leq y - 1, & & z' = z\}
\end{array}$$

Note that $\mathcal{Q}_i$ are convex polyhedra. A transition from a state $\bar{x}$ to $\bar{x}'$ is possible iff $(\bar{x}, \bar{x}')$ is a point in some path $\mathcal{Q}_i$. We remark that our results hold for arbitrarily-complex control-flow graphs (CFGs), we prefer to use the loop setting for clarity.

A popular tool for proving the termination of such loops is *linear ranking functions* (LRFs). An LRF is a function $\rho(x_1, \ldots, x_n) = a_1 x_1 + \cdots + a_n x_n + a_0$ such that any transition $(\bar{x}, \bar{x}')$ satisfies (i) $\rho(\bar{x}) \geq 0$; and (ii) $\rho(\bar{x}) - \rho(\bar{x}') \geq 1$. E.g., $\rho(x, y, z) = x$ is an LRF for a loop that consists of only $\mathcal{Q}_1$ and $\mathcal{Q}_2$ above, $\rho(x, y, z) = y$ is an LRF for $\mathcal{Q}_3$ and $\mathcal{Q}_4$, and $\rho(x, y, z) = z$ is an LRF for $\mathcal{Q}_2$ and $\mathcal{Q}_3$. However, there is no LRF that satisfies the above conditions for all paths $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$. An algorithm to find an LRF using linear programming (LP) has been found by multiple researchers in different places and times and in some alternative versions [1,9,13,21,23,26]. Since LP has a polynomial-time complexity, most of these methods yield polynomial-time algorithms. These algorithms are complete for loops with rational-valued variables, but not with integer-valued variables. Indeed, [3] shows loops that have LRFs over the integers but do not even terminate over the rationals. In a previous work [3] we considered the integer setting, where complete algorithms were proposed and a complexity classification was proved: to decide whether an LRF exists is *coNP-complete*.

LRFs do not suffice for all loops (e.g., the 4-path loop above), and thus, a natural question is what to do when an LRF does not exist; and a natural answer is to try a richer class of ranking functions. Of particular importance is the class of *lexicographic-linear ranking functions* (LLRFs). An LLRF is a d-tuple of affine-linear functions, $\langle \rho_1, \ldots, \rho_d \rangle$, required to descend lexicographically. Interestingly, Alan Turing's early demonstration [28] of how to verify a program used an LLRF for the termination proof. *Algorithms* to find LLRFs for linear-constraint loops (or CFGs) can use LP techniques, extending the work on LRFs. Alias et al. [1] extended the polynomial-time LRF algorithm to LLRFs and gave a complete solution for CFGs. As for LRFs, the solution is incomplete for integer data, and in [3] we established for LLRFs over the integers results that parallel those for LRFs, in particular, to decide whether an LLRF exists is *coNP-complete*.

Interestingly, when trying to define the requirements from a numeric "lexicographic ranking function" (corresponding to the conditions (i) and (ii) on an LRF, above), different researchers had come up with different definitions. In particular, the definition in [1] is more restrictive than the definition in [3]. Furthermore, an important paper [4] on LLRF generation that preceded both works gave yet a different definition. We give the precise definitions in Sect. 2; for the purpose of introduction, let us focus on the LLRFs of [4] (henceforth, BMS-LLRFs, after the authors), and illustrate the definition by an example.

Consider the above loop defined by $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$. A possible BMS-LLRF for this loop is $\rho(x, y, z) = \langle x, y \rangle$. The justification is this: in $\mathcal{Q}_1$ and $\mathcal{Q}_2$, the function

$\rho_1(x, y) = x$ is ranking (non-negative and decreasing by at least 1). In $\mathcal{Q}_3$ and $\mathcal{Q}_4$, $\rho_2(x, y) = y$ is ranking, while ρ_1 is non-increasing. This is true over the rationals and *a fortiori* over the integers. The following points are important: (1) for each path we have an LRF, which is one of the components of the BMS-LLRF; and (2) previous (lower-numbered) components are only required to be non-increasing on that path. Note that this LLRF does not satisfy the requirements of [1] or [3].

The goal of this paper is to understand the *computational complexity* of some problems related to BMS-LLRFs, starting with the most basic problem, whether a given loop has such LLRF. We note that [4] does not provide an answer, as a consequence of attempting to solve a much harder problem—they consider a loop given with a precondition and search for a BMS-LLRF together with a supporting linear invariant. We do not know if this problem is even decidable when parameters like the number of constraints in the invariants are not fixed in advance (when they are, the approach of [4] is complete, but only over the reals, and at a high computational cost – even without a precondition).

We consider the complexity of finding a BMS-LLRF for a given loop, without preconditions. We prove that this can be done in polynomial time when the loop is interpreted over the rationals, while over the integers, deciding the existence of a BMS-LLRF is coNP-complete. An exponential-time synthesis algorithm is also given. These results are similar to those obtained for the previously studied classes of LLRFs [3], but are shown for the first time for BMS-LLRFs.

Next, we consider the number of components d in a BMS-LLRF $\langle \rho_1, \ldots, \rho_d \rangle$. This number is informally called the *dimension* of the function. It is interesting for several reasons: An upper bound on the dimension is useful for fixing the template in the constraint-solving approach, and plays a role in analyzing the complexity of corresponding algorithms. In addition, an LLRF can be used to infer bounds on the number of iterations [1]; assuming linear bounds on individual variables, a polynomial bound of degree d is clearly implied, which motivates the desire to minimize the dimension, to obtain tight bounds. A smaller dimension also means better results when LLRFs are used to guide parallelization [14].

Importantly, the algorithms of Alias et al. [1] and Ben-Amram and Genaim [3] are optimal w.r.t. the dimension, i.e., they synthesize LLRFs of minimal dimension for the respective classes. We note that it is possible for a loop to have LLRFs of all three classes but such that the minimal dimension is different in each (see Sect. 4). We also note that, unlike the case for the previous classes, our synthesis algorithm for BMS-LLRFs is *not* guaranteed to produce a function of minimal dimension. This leads us to ask: (1) what is the best *a priori* bound on the dimension, in terms of the number of variables and paths; and (2) how difficult it is to find an LLRF of minimal dimension. As a relaxation of this optimization problem, we can pose the problem of finding an LLRF that satisfies a given bound on the dimension. Our results are summarized in Table 1. There is a striking difference of BMS-LLRFs from other classes w.r.t. to the minimum dimension problem: the complexity jumps from PTIME (resp. coNP-complete) to NPC (resp. Σ_2^P-complete) over rationals (resp. integers). This holds for any fixed dimension larger than one (dimension one is an LRF).

LLRF type	Dimension bound	Existence		Fixed dimension	
		over $\mathbb{Q}$	over $\mathbb{Z}$	over $\mathbb{Q}$	over $\mathbb{Z}$
ADFG [1]	$\min(n,k)$	PTIME	coNP-complete	PTIME	coNP-complete
BG [3]	n	PTIME	coNP-complete	PTIME	coNP-complete
BMS [4]	k	PTIME	coNP-complete	NP-complete	Σ_2^P-complete

Table 1. Summary of results, considering a loop of k paths over n variables. Those in the third row are new, the others are from previous works or follow by minor variations.

2 Preliminaries

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 $A\mathbf{x} \leq \mathbf{b}$. We use calligraphic letters, such as $\mathcal{P}$ and $\mathcal{Q}$ to denote polyhedra. 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 $I(\mathcal{P})$. It is known that $\mathcal{P}_I$ is also a polyhedron. An *integer polyhedron* is a polyhedron $\mathcal{P}$ such that $\mathcal{P} = \mathcal{P}_I$. We also say that $\mathcal{P}$ is *integral*.

Multipath Linear-Constraint Loops. A *multipath* linear-constraint loop (MLC loop) with k paths has the form: $\bigvee_{i=1}^{k} A_i \binom{\mathbf{x}}{\mathbf{x}'} \leq \mathbf{c}_i$ 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 $q > 0$, $A_i \in \mathbb{Q}^{q \times 2n}$, $\mathbf{c}_i \in \mathbb{Q}^q$. Each path $A_i \binom{\mathbf{x}}{\mathbf{x}'} \leq \mathbf{c}_i$ is called an *abstract transition*. The loop is a *rational loop* if $\mathbf{x}$ and $\mathbf{x}'$ range over $\mathbb{Q}^n$, and it is an *integer loop* if they range over $\mathbb{Z}^n$. 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 for some $1 \leq i \leq k$, $\binom{\mathbf{x}}{\mathbf{x}'}$ satisfies the i-th abstract transition. In such case we say that $\mathbf{x}$ is an enabled state. We use $\mathbf{x}''$ as a shorthand for a transition $\binom{\mathbf{x}}{\mathbf{x}'}$, and consider it as a point in $\mathbb{Q}^{2n}$. The set of transitions satisfying a particular abstract transition is a polyhedron in $\mathbb{Q}^{2n}$, denoted $\mathcal{Q}_i$, namely $A_i\mathbf{x}'' \leq \mathbf{c}_i$. In our work it is convenient to represent an MLC loop by its transition polyhedra $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, which we often write with explicit equalities and inequalities. These are sometimes referred to as the *paths* of the multipath loop.

Ranking Functions. An affine linear function $\rho : \mathbb{Q}^n \mapsto \mathbb{Q}$ is of the form $\rho(\mathbf{x}) = \vec{\lambda} \cdot \mathbf{x} + \lambda_0$ where $\vec{\lambda} \in \mathbb{Q}^n$ and $\lambda_0 \in \mathbb{Q}$. We define $\Delta\rho : \mathbb{Q}^{2n} \mapsto \mathbb{Q}$ as $\Delta\rho(\mathbf{x}'') = \rho(\mathbf{x}) - \rho(\mathbf{x}')$. Given a set $T \subseteq \mathbb{Q}^{2n}$, representing transitions, we say that ρ is an LRF for T if for every $\mathbf{x}'' \in T$ we have (i) $\rho(\mathbf{x}) \geq 0$; and (ii) $\Delta\rho(\mathbf{x}'') \geq 1$. We say that ρ is an LRF for a rational (resp. integer) loop, specified by $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, when it is an LRF for $\bigcup_{i=1}^{k} \mathcal{Q}_i$ (resp. $\bigcup_{i=1}^{k} I(\mathcal{Q}_i)$). For a rational loop, there is a polynomial-time algorithm to either find an LRF or determine that none exists [23]. Its essence is that using *Farkas' Lemma* [25, p. 93], it is possible to set up an LP problem whose feasibility is equivalent to the existence of ρ that satisfies (i) and (ii) over $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$.

A *d-dimensional affine function* $\tau : \mathbb{Q}^n \to \mathbb{Q}^d$ is expressed by a d-tuple $\tau = \langle \rho_1, \ldots, \rho_d \rangle$, where each component $\rho_i : \mathbb{Q}^n \to \mathbb{Q}$ is an affine linear function. The number d is informally called the *dimension* of τ. Next we define when such a function is BMS-LLRF [4] for a given rational or integer MLC loop. We then compare with ADFG-LLRFs (due to [1]) and BG-LLRFs (due to [3]).

Definition 1 (BMS-LLRF). *Given k sets of transitions $T_1, \ldots, T_k \subseteq \mathbb{Q}^{2n}$, we say that $\tau = \langle \rho_1, \ldots, \rho_d \rangle$ is a* BMS-LLRF *for $T_1, \ldots, T_k$ iff for every $1 \leq \ell \leq k$ there is $1 \leq i \leq d$ such that the following hold for any $\mathbf{x}'' \in T_\ell$:*

$$\forall j < i \;.\; \Delta\rho_j(\mathbf{x}'') \geq 0\,, \tag{1}$$
$$\rho_i(\mathbf{x}) \geq 0\,, \tag{2}$$
$$\Delta\rho_i(\mathbf{x}'') \geq 1\,. \tag{3}$$

We say that T_ℓ is ranked by ρ_i.

We say that τ is a BMS-LLRF for a rational (resp. integer) loop, specified by $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, when it is a BMS-LLRF for $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ (resp. $I(\mathcal{Q}_1), \cdots, I(\mathcal{Q}_k)$). It is easy to see that the existence of a BMS-LLRF implies termination.

Definition 2 (BG-LLRF). *Given a set of transitions $T \subseteq \mathbb{Q}^{2n}$, we say that $\tau = \langle \rho_1, \ldots, \rho_d \rangle$ is a* BG-LLRF *for T iff for every $\mathbf{x}'' \in T$ there is $1 \leq i \leq d$ such that the following hold:*

$$\forall j < i \;.\; \Delta\rho_j(\mathbf{x}'') \geq 0\,, \tag{4}$$
$$\forall j \leq i \;.\; \quad \rho_j(\mathbf{x}) \geq 0\,, \tag{5}$$
$$\Delta\rho_i(\mathbf{x}'') \geq 1\,. \tag{6}$$

We say that $\mathbf{x}$ is ranked by ρ_i.

We say that τ is a BG-LLRF for a rational (resp. integer) loop, specified by $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, when it is a BG-LLRF for $\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_k$ (resp. $I(\mathcal{Q}_1) \cup \cdots \cup I(\mathcal{Q}_k)$). It is easy to see that the existence of a BG-LLRF implies termination.

Note the differences between the definitions: in one sense, BG-LLRFs are more flexible because of the different quantification — for every transition $\mathbf{x}''$ there has to be a component ρ_i that ranks it, but i may differ for different $\mathbf{x}''$, whereas in BMS-LLRFs, all transitions that belong to a certain T_ℓ have to be ranked by the same component. In another sense, BMS-LLRFs are more flexible because components ρ_j with $j < i$ can be negative (compare (2) with (5)). Thus, there are loops that have a BMS-LLRF and do not have a BG-LLRF (see loop in Sect. 1); and vice versa (see [3, Ex. 2.12]). A third type of LLRFs is attributed to [1], hence we refer to it as ADFG-LLRF. It is similar to BG-LLRFs but requires all components to be non-negative in every enabled state. That is, condition (5) is strengthened. Interestingly, the completeness proof in [1] shows that the above-mentioned flexibility of BG-LLRFs adds no power in this case; therefore, ADFG-LLRFs are a special case of both BG-LLRFs and BMS-LLRFs.

The decision problem *Existence of a* BMS-LLRF deals with deciding whether a given MLC loop admits a BMS-LLRF, we denote it by BMS-LexLinRF($\mathbb{Q}$) and BMS-LexLinRF($\mathbb{Z}$) for rational and integer loops respectively. The corresponding decision problems for ADFG- and BG-LLRFs are solved in [1] and [3], respectively, over the rationals; the case of integers is only addressed in [3] for BG-LLRFs, but the complexity results apply to ADFG-LLRFs as well.

Algorithm 1: Synthesizing BMS-LLRFs

```
   LLRFSYN(⟨Q_1, ..., Q_k⟩)
   begin
1      if ⟨Q_1, ..., Q_k⟩ are all empty then return nil if Q_1, ..., Q_k has a
       BMS-QLRF ρ then
3          ∀1 ≤ i ≤ k. Q'_i := ∅ if Q_i is ranked by ρ, otherwise Q'_i = Q_i
4          τ ← LLRFSYN(⟨Q'_1, ..., Q'_k⟩)
5          if τ ≠ NONE then return ρ::τ
6      return NONE
```

3 Synthesis of BMS-LLRFs

In this section we describe a complete algorithm for synthesizing BMS-LLRFs for rational and integer MLC loops; and show that the decision problems BMS-LexLinRF($\mathbb{Q}$) and BMS-LexLinRF($\mathbb{Z}$) are PTIME and coNP-complete, respectively. We assume a given MLC loop $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ where each $\mathcal{Q}_i$ is given as a set of linear constraints, over $2n$ variables (n variables and n primed variables).

Definition 3. *Let $T_1, \ldots, T_k$ be sets of transitions such that $T_i \subseteq \mathbb{Q}^{2n}$. We say that an affine linear function ρ is a* BMS *quasi-*LRF *(*BMS-QLRF *for short) for $T_1, \ldots, T_k$ if every transition $\mathbf{x}'' \in T_1 \cup \cdots \cup T_k$ satisfies $\Delta\rho(\mathbf{x}'') \geq 0$, and for at least one T_ℓ, ρ is an* LRF *(such T_ℓ is said to be ranked by ρ).*

Example 1. The following are BMS-QLRFs for the loop consisting of $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$ presented in Sect. 1: $f_1(x, y, z)=x$, which ranks $\{\mathcal{Q}_1, \mathcal{Q}_2\}$; $f_2(x, y, z)=y$ which ranks $\{\mathcal{Q}_3, \mathcal{Q}_4\}$; and $f_3(x, y, z)=z$ which ranks $\{\mathcal{Q}_2, \mathcal{Q}_3\}$.

Lemma 1. *There is a polynomial-time algorithm that finds a* BMS-QLRF *ρ, if there is any, for $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$.*

Proof. The algorithm iterates over the paths $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$. In the i-th iteration it checks if there is an LRF ρ for $\mathcal{Q}_i$ that is non-increasing for all other paths, stopping if it finds one. The algorithm makes at most k iterations. Each iteration can be implemented in polynomial time using Farkas' Lemma (as in [23]). □

Our procedure for synthesizing BMS-LLRFs is depicted in Alg. 1. In each iteration (i.e., call to `LLRFSYN`): it finds a BMS-QLRF ρ for the current paths (Line 2); it eliminates all paths that are ranked by ρ (Line 3); and calls recursively to handle the remaining paths (Line 4). The algorithm stops when all paths are ranked (Line 1), or when it does not find a BMS-QLRF (Line 6).

Example 2. Consider the MLC loop example in Sect. 1. Procedure `LLRFSYN` is first applied to $\langle \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4 \rangle$, and at Line 2 we can choose the BMS-QLRF x which ranks $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Hence these are eliminated at Line 3, and at Line 4 `LLRFSYN` is applied recursively to $\langle \emptyset, \emptyset, \mathcal{Q}_3, \mathcal{Q}_4 \rangle$. Then at Line 2 we can choose the BMS-QLRF y which ranks $\mathcal{Q}_3$ and $\mathcal{Q}_4$. The next recursive call receives empty polyhedra, and thus the check at Line 1 succeeds and the algorithm returns $\langle x, y \rangle$.

Lemma 2. *If* `LLRFSYN`$(\langle \mathcal{Q}_1, \ldots, \mathcal{Q}_k \rangle)$ *returns* τ *different from* NONE*, then* τ *is a* BMS-LLRF *for the rational loop* $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$*.*

The proof of the above lemma is straightforward. Thus, Alg. 1 is a sound algorithm for BMS-LLRFs. The following proposition shows completeness.

Proposition 1. *There is a* BMS-LLRF *for* $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ *if and only if every subset of* $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\}$ *has a* BMS-QLRF*.*

Proof. The "if" direction is implied by the `LLRFSYN` procedure, in such case it will find a BMS-LLRF. For the "only if" direction, let $\tau = \langle \rho_1, \ldots, \rho_d \rangle$ be a BMS-LLRF for $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, and let $\mathcal{Q}_{\ell_1}, \ldots, \mathcal{Q}_{\ell_j}$ be an arbitrary subset of the loop's paths. Since τ is a BMS-LLRF for $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$, each $\mathcal{Q}_{\ell_i}$ is ranked by some ρ_{l_i}. Let $l = \min\{l_1, \ldots, l_j\}$, then ρ_l is a BMS-QLRF for $\mathcal{Q}_{\ell_1}, \ldots, \mathcal{Q}_{\ell_j}$. □

Lemma 3. *Procedure* `LLRFSYN` *can be implemented in polynomial time.*

Proof. Procedure `LLRFSYN` makes at most k steps (since at least one path is eliminated in every step). Further, all steps are elementary except checking for a BMS-QLRF which can be done in polynomial time as stated by Lemma 1. □

Corollary 1. BMS-LEXLINRF$(\mathbb{Q}) \in$ *PTIME.*

So far we have considered only rational loops, next we consider integer loops.

Lemma 4. *There is a complete algorithm for synthesizing a* BMS-QLRF *for* $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$*.*

Proof. The algorithm computes the integer hull $\mathcal{Q}_{1I}, \ldots, \mathcal{Q}_{kI}$, and then proceeds as in the rational case (Lemma 1). Correctness follows from the fact that for integral polyhedra the implied inequalities over the rationals and integers coincide, i.e., $\mathcal{Q}_{1I}, \ldots, \mathcal{Q}_{kI}$ and $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ have the same BMS-QLRFs. □

Lemma 5. *When procedure* `LLRFSYN` *is applied to the integer hulls* $\mathcal{Q}_{1I}, \ldots, \mathcal{Q}_{kI}$*, it finds a* BMS-LLRF *for* $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$*, if one exists.*

Proof. Soundness follows from the fact that $\mathcal{Q}_I$ contains $I(\mathcal{Q})$; for completeness, note that: (i) Prop. 1 holds also for integer loops; and (ii) Line 3 of `LLRFSYN` does not change the transition polyhedra, it only eliminates some, which means that they remain integral throughout the recursive calls. Thus, in each iteration the check at Line 2 is complete (see Lemma 4). □

In the general case this procedure has an exponential time complexity since computing the integer hull requires an exponential time. However, for special cases in which the integer hull can be computed in polynomial time [3, Sect. 4] it has polynomial time complexity. The following lemma implies (assuming P≠NP) that the exponential time complexity is unavoidable in general.

Theorem 1. BMS-LEXLINRF$(\mathbb{Z})$ *is a coNP-complete problem.*

Proof. The coNP-hardness follows from the reduction in [3, Sect. 3.1], since it constructs a loop that either does not terminate or has an LRF. The inclusion in coNP is based on arguments similar to those in [3, Sect. 5]; briefly, we use the generator representation of the transition polyhedra to construct a polynomial-size witness against existence of an LLRF (see App. A). □

4 The Dimension of BMS-LLRFs

Ben-Amram and Genaim [3, Cor. 5.12, p. 32] showed that if a given MLC loop has a BG-LLRF, then it has one of dimension at most n, the dimension of the state space. The same proof can be used to bound the dimension of ADFG-LLRFs by n as well. Hence for ADFG-LLRFs the bound $\min(n, k)$ holds (k is the number of paths), due to the fact that associating LLRF components with paths is no loss of generality for ADFG-LLRFs [1]. In the case of BMS-LLRFs, the bound k clearly holds, and the next example shows that it is tight.

Example 3. Define an MLC loop $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ for some $k > 0$, over variables x, y, where each $\mathcal{Q}_i = \{x' \leq x,\ x' + i \cdot y' \leq x + i \cdot y - 1,\ x + i \cdot y \geq 0\}$. Define $f_i(x, y) = x + i \cdot y$. It is easy to check that (i) f_i is an LRF for $\mathcal{Q}_i$, and is non-increasing for any $\mathcal{Q}_j$ with $i < j \leq k$; and (ii) there are no distinct $\mathcal{Q}_i$ and $\mathcal{Q}_j$ that have a common LRF. From (i) it follows that $\langle f_1, \ldots, f_k \rangle$ is a BMS-LLRF for this loop, and from (ii) it follows that any BMS-LLRF must have (at least) dimension k, since different paths cannot be ranked by the same component. We remark that this loop has no BG-LLRF (hence, also no ADFG-LLRF).

The above discussion emphasizes the difference between the various definitions of LLRFs, when considering the dimension. The next example emphasizes this difference further, it shows that there are loops, having LLRFs of all three kinds, for which the minimal dimension is different according to each definition. This also means that the implied bounds on the number of iterations (assuming, for simplicity, that all variables have the same upper bound) are different.

Example 4. Consider an MLC loop specified by the following paths

$$\mathcal{Q}_1 = \left\{ \begin{array}{llllllll} r \geq 0, & & t \geq 0, & x \geq 0, & & z \geq 0, & w \geq 0, \\ r' < r, & & t' < t, & & & & \end{array} \right\}$$

$$\mathcal{Q}_2 = \left\{ \begin{array}{llllllll} r \geq 0, & s \geq 0, & t \geq 0, & x \geq 0, & & z \geq 0, & w \geq 0, \\ r' = r, & s' < s, & t' < t, & & & & \end{array} \right\}$$

$$\mathcal{Q}_3 = \left\{ \begin{array}{llllllll} r \geq 0, & s \geq 0, & t' = t & x \geq 0, & & z \geq 0, & w \geq 0, \\ r' = r, & s' = s, & & x' < x, & & & \end{array} \right\}$$

$$\mathcal{Q}_4 = \left\{ \begin{array}{llllllll} r \geq 0, & s \geq 0, & t' = t & x \geq 0, & y \geq 0, & z \geq 0, & w \geq 0, \\ r' = r, & s' = s, & & x' = x, & y' < y, & z' < z, & \end{array} \right\}$$

$$\mathcal{Q}_5 = \left\{ \begin{array}{llllllll} r \geq 0, & s \geq 0, & t' = t & x \geq 0, & y \geq 0, & z \geq 0, & w \geq 0, \\ r' = r, & s' = s, & & x' = x, & y' < y, & z' = z, & w' < w \end{array} \right\}$$

where, for readability, we use $<$ for the relation "smaller at least by 1". This loop has the BMS-LLRF $\langle t, x, y \rangle$, which is neither a BG-LLRF or ADFG-LLRF because t is not lower-bounded on all the paths. Its shortest BG-LLRF is of dimension 4, e.g., $\langle r, s, x, y \rangle$, which is not an ADFG-LLRF because y is not lower-bounded on all the paths. Its shortest ADFG-LLRF is of dimension 5, e.g., $\langle r, s, x, z, w \rangle$. This reasoning is valid for both integer and rational variables.

Next, we consider the problem of minimal dimension. We ask (1) whether our algorithms return an LLRF with minimal dimension; and (2) what do we gain (or lose?) in terms of computational tractability if we fix a bound on the

dimension in advance. Importantly, the algorithms of [1,3] are optimal w.r.t. the dimension, i.e., they synthesize LLRFs of minimal dimension. In both cases the optimal result is obtained by a greedy algorithm, that constructs the LLRF by adding one dimension at a time, taking care in each iteration to rank as many transitions as possible. The next example shows that a greedy choice in Alg. 1 fails to guarantee optimality, for both rational and integer loops. Intuitively, the greedy approach worked in [1,3] because the classes of quasi-LRFs used to construact LLRFs are closed under conic combinations, so there is always an optimal choice that dominates all others. This is not true for BMS-QLRFs.

Example 5. Consider the MLC loop of Sect. 1. If at Line 2 Alg. 1 we seek a BMS-QLRF that ranks a maximal number of the paths, we can use any of those derived in Ex. 1: $f_1 = x$; $f_2 = y$; or $f_3 = z$. However, these alternatives lead to BMS-LLRFs of different dimensions: (i) choose f_1 to rank $\{\mathcal{Q}_1, \mathcal{Q}_2\}$, and then f_2 to rank $\{\mathcal{Q}_3, \mathcal{Q}_4\}$. (ii) choose f_2 to rank $\{\mathcal{Q}_3, \mathcal{Q}_4\}$, and then f_1 to rank $\{\mathcal{Q}_1, \mathcal{Q}_2\}$. (iii) choose f_3 to rank $\{\mathcal{Q}_2, \mathcal{Q}_3\}$, but then there is no single function that ranks $\{\mathcal{Q}_1, \mathcal{Q}_4\}$. Take f_1 to rank $\mathcal{Q}_1$ and then f_2 to rank $\mathcal{Q}_4$. The dimension of the BMS-LLRF in the first two cases is 2, and in the last one it is 3.

Since Alg. 1 is not guaranteed to find a BMS-LLRF of minimal dimension, it is natural to ask *how hard is the problem of finding a* BMS-LLRF *of minimal dimension?* This can be posed as a decision problem: *does a given MLC loop have a* BMS-LLRF *with dimension at most d*? This decision problem is denoted by BMS-LexLinRF$(d, \mathbb{Q})$ and BMS-LexLinRF$(d, \mathbb{Z})$ for rational and integer loops respectively. Note that d is a constant, however, it will be clear that accepting d as an input does not change the complexity class of these problems. Also note that for $d = 1$ it is just the LRF problem. Similar problems can be formulated for ADFG- and BG-LLRFs, of course. In these two settings, the imposition of a dimension bound does not change the complexity class.

Theorem 2. *Given a rational MLC loop, and $d \geq 1$, it is possible to determine in polynomial time if there is an* ADFG-LLRF *(resp.* BG-LLRF*s) for the loop of dimension at most d. For integer MLC loops, the problem is coNP-complete.*

Proof. The case of rational loops is straightforward since the corresponding synthesis algorithms find LLRFs with minimal dimension, and are in PTIME. The integer case follows easily from the techniques of [3] (see App. B). □

5 Complexity of BMS-LexLinRF$(d, \mathbb{Q})$

In this section we show that BMS-LexLinRF$(d, \mathbb{Q})$ is NP-complete.

Theorem 3. *For $d \geq 2$,* BMS-LexLinRF$(d, \mathbb{Q})$ *is an NP-complete problem.*

For inclusion in NP, a non-deterministic algorithm for the problem works as follows. First, it *guesses* a partition of $\{1, \ldots, k\}$ into d sets $J_1, \ldots, J_d$, of which some may be empty (we can assume they are last). Then it proceeds as in Alg. 1

but insists that the paths indexed by J_r be ranked at the r-th iteration. This may fail, and then the algorithm rejects. If a BMS-LLRF of dimension at most d exists, there will be an accepting computation.

For NP-hardness we reduce from the NP-complete problem *d-Colorability of 3-Uniform Hypergraphs* [20,22]. An instance of this problem is a set H of m sets $F_1, \ldots, F_m$ (hyperedges, or "faces"), where each F_i includes exactly 3 elements from a set of vertices $V = \{1, \ldots, n\}$, and we are asked whether we can choose a color (out of d colors) for each vertex such that every face is not monocolored.

We construct a rational MLC loop in $3m$ variables and n paths. The variables are indexed by vertices and faces: variable $x_{i,j}$ corresponds to $i \in F_j \in H$. For each vertex $1 \leq i \leq n$ we define $\mathcal{Q}_i$ as a conjunction of the following:

$$\sum_{k\colon\, i \in F_k} x_{i,k} - \sum_{k\colon\, i \in F_k} x'_{i,k} \geq 1 \tag{7}$$

$$\sum_{k\colon\, j \in F_k} x_{j,k} - \sum_{k\colon\, j \in F_k} x'_{j,k} \geq 0 \quad \text{for all vertex } j \neq i \tag{8}$$

$$x_{i,k} \geq 0 \quad \text{for all face } F_k \text{ s.t. } i \in F_k \tag{9}$$

$$x_{j,k} \geq 0 \quad \text{for all vertex } j \text{ and face } F_k \text{ s.t. } j \in F_k \wedge i \notin F_k \tag{10}$$

$$x_{i,k} + x_{j,k} \geq 0 \quad \text{for all vertex } j \neq i \text{ and face } F_k \text{ s.t. } i, j \in F_k \tag{11}$$

We claim that a rational loop that consists of these n paths has a BMS-LLRF of dimension d iff there is a valid d-coloring for the vertices V.

Assume given a d-coloring, namely a division of the vertices in d disjoint sets $V = C_1 \cup \cdots \cup C_d$, such that the vertices of each C_i are assigned the same color. We construct a BMS-LLRF $\langle g_1, \ldots, g_d \rangle$ such that g_ℓ ranks all paths Q_i with $i \in C_\ell$. We assume that each C_ℓ is non-empty (otherwise we let $g_\ell(\mathbf{x}) = 0$).

We start with C_1. For each $F_k \in H$, define a function f_k as follows: if $F_k \cap C_1 = \emptyset$ we let $f_k(\mathbf{x}) = 0$; if $F_k \cap C_1 = \{i\}$ we let $f_k(\mathbf{x}) = x_{i,k}$; and if $F_k \cap C_1 = \{i, j\}$ we let $f_k(\mathbf{x}) = x_{i,k} + x_{j,k}$. We claim that $g_1(\mathbf{x}) = \sum_k f_k$ is a BMS-QLRF for $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$ that ranks all paths $\mathcal{Q}_i$ with $i \in C_1$, which we justify as follows:

1. g_1 is non-increasing on all $\mathcal{Q}_j$, and decreasing for each $\mathcal{Q}_i$ with $i \in C_1$. To see this, rewrite $g(\mathbf{x})$ as $\sum_{\iota \in C_1} \sum_{k\colon\, \iota \in F_k} x_{\iota,k}$. As each inner sum is non-increasing by (7,8), we conclude that g_1 is non-increasing on all paths. Moreover, for $i \in C_1$, the sum $\sum_{k\colon\, i \in F_k} x_{i,k}$ appears in g_1 and is decreasing according to (7), thus g_1 is decreasing for each $\mathcal{Q}_i$ with $i \in C_1$.
2. g_1 is non-negative for all $\mathcal{Q}_i$ with $i \in C_1$, because all f_k are non-negative on these paths. To see this, pick an arbitrary $i \in C_1$ and an arbitrary face F_k: if $i \in F_k$, and it is the only vertex from C_1 in F_k, then $f_k(\mathbf{x}) = x_{i,k}$ is non-negative on $\mathcal{Q}_i$ by (9); if $i \in F_k$ but there is another vertex $j \in C_1$ in F_k, then $f_k(\mathbf{x}) = x_{i,k} + x_{j,k}$ is non-negative on $\mathcal{Q}_i$ by (11); if $i \notin F_k$, then for any $j \in F_k$ we have $x_{j,k} \geq 0$ by (10), and then f_k is non-negative since it is a sum of such variables. Note that g_1 can be negative for $\mathcal{Q}_j$ with $j \notin C_1$.

Similarly, we construct BMS-QLRFs $g_2, \ldots, g_d$ such that g_ℓ ranks $\mathcal{Q}_i$ for $i \in C_\ell$. Clearly $\langle g_1, \ldots, g_d \rangle$ is a BMS-LLRF for this loop.

Now suppose we have a BMS-LLRF of dimension d; we analyze what paths $\mathcal{Q}_i$ can be associated with each component, and show that for any face F_k, the three paths that are indexed by its vertices, i.e., $\mathcal{Q}_i$ for $i \in F_k$, cannot be all associated with the same component. Which clearly yields a d-coloring.

Suppose that for some face $F_k = \{i_1, i_2, i_3\}$, the paths $\mathcal{Q}_{i_1}, \mathcal{Q}_{i_2}$ and $\mathcal{Q}_{i_3}$ are associated with the same component, i.e., all ranked by the same function, say g. Thus $\Delta g(\mathbf{x}'') \geq 1$ must be implied by the constraints of $\mathcal{Q}_{i_1}, \mathcal{Q}_{i_2}$ and $\mathcal{Q}_{i_3}$, independently. Now since, in each path, the only constraint with a non-zero free coefficient is (7), it follows that the coefficients of variables $x_{i_1,k}$, $x_{i_2,k}$ and $x_{i_3,k}$ in $g(\mathbf{x})$ are positive, i.e., $g(\mathbf{x}) = a_1 \cdot x_{i_1,k} + a_2 \cdot x_{i_2,k} + a_3 \cdot x_{i_3,k} + h(\mathbf{x})$ where $h(\mathbf{x})$ is a combination of other variables, and $a_1, a_2, a_3 > 0$. Similarly, $g(\mathbf{x}) \geq 0$ must be implied by the constraints of each of three paths independently. For this to hold, g must be a positive linear combination of functions constrained to be non-negative by these paths, and do not involve primed variables. Now consider variables $x_{i_1,k}$, $x_{i_2,k}$ and $x_{i_3,k}$, and note that they participate only in the following constraints in $\mathcal{Q}_{i_1}$ (left), $\mathcal{Q}_{i_2}$ (middle) and $\mathcal{Q}_{i_3}$ (right):

$$\begin{array}{r} x_{i_1,k} \geq 0 \\ x_{i_1,k} + x_{i_2,k} \geq 0 \\ x_{i_1,k} + x_{i_3,k} \geq 0 \end{array} \qquad \begin{array}{r} x_{i_2,k} \geq 0 \\ x_{i_1,k} + x_{i_2,k} \geq 0 \\ x_{i_2,k} + x_{i_3,k} \geq 0 \end{array} \qquad \begin{array}{r} x_{i_3,k} \geq 0 \\ x_{i_2,k} + x_{i_3,k} \geq 0 \\ x_{i_1,k} + x_{i_3,k} \geq 0 \end{array}$$

This means that the corresponding coefficients in g, i.e., $\bar{a} = (a_1\ a_2\ a_3)$, must be equal to linear combinations of the corresponding coefficients in the above constraints. Namely, there exist $b_1, \ldots, b_9 \geq 0$ such that

$$\bar{a} = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \qquad \bar{a} = \begin{pmatrix} b_4 & b_5 & b_6 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \qquad \bar{a} = \begin{pmatrix} b_7 & b_8 & b_9 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

From these nine equations, and the constraints $b_i \geq 0$ for all i, we necessarily get $a_1 = a_2 = a_3 = 0$, which contradicts $a_1, a_2, a_3 > 0$ as we concluded before, and thus paths corresponding to $\{i_1, i_2, i_3\}$ of F_k cannot be all associated with the same component. This concludes the proof of Th. 3.

6 Complexity of BMS-LexLinRF($d, \mathbb{Z}$)

In this section we turn to the problem BMS-LexLinRF($d, \mathbb{Z}$), and show that it is harder than BMS-LexLinRF($d, \mathbb{Q}$), specifically, it is Σ_2^P-complete. The class Σ_2^P is the class of decision problems that can be solved by a standard, non-deterministic computational model in polynomial time assuming access to an oracle for an NP-complete problem. I.e., $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$. This class contains both NP and coNP, and is likely to differ from them both (this is an open problem).

Theorem 4. *For $d \geq 2$,* BMS-LexLinRF($d, \mathbb{Z}$) *is a Σ_2^P-complete problem.*

The rest of this section proves Th. 4. For inclusion in Σ_2^P we use a non-deterministic procedure as in the proof of Th. 3. Note that the procedure needs to find (or check for existence of) BMS-QLRFs over the integers, so it needs a coNP oracle. For Σ_2^P-hardness we reduce from the canonical Σ_2^P-complete problem (follows from [27, Th. 4.1]): evaluation of sentences of the form

$$\exists X_1 \ldots X_n\ \forall X_{n+1} \ldots X_{2n}\ \neg\phi(X_1, \ldots, X_{2n}) \qquad (\star)$$

where the variables X_i are Boolean and the formula ϕ is in 3CNF form. Thus, ϕ is given as a collection of m clauses, $C_1, \ldots, C_m$, each clause C_i consisting of three literals $L_i^j \in \{X_1, \ldots, X_{2n},\ \neg X_1, \ldots, \neg X_{2n}\}$, $1 \leq j \leq 3$. The reduction is first done for $d = 2$, and later extended to $d > 2$ as well.

Let us first explain a well-known approach for reducing satisfiability of a Boolean formula ϕ to satisfiability of integer linear constraints. We first associate each literal L_i^j with an integer variables $x_{i,j}$. Note that the same Boolean variable (or its complement) might be associated with several constraint variables. Let C be the set of (1) all conflicting pairs, that is, pairs $((i,j),(r,s))$ such that L_i^j is the complement of L_r^s; and (2) pairs $((i,j),(i,j'))$ with $1 \leq j < j' \leq 3$, i.e., pairs of literals that appear in the same clause. We let $\mathcal{F}$ be a conjunction of the constraints: $x_{i,j} + x_{r,s} \leq 1$ for each $((i,j),(r,s)) \in C$; and $0 \leq x_{i,j} \leq 1$ for each $1 \leq i \leq m$ and $1 \leq j \leq 3$. An assignment for $x_{i,j}$ that satisfies $\mathcal{F}$ is called a *non-conflicting assignment*, since if two variables correspond to conflicting literals (or to literals of the same clause) they cannot be assigned 1 at the same time. The next Lemma relates integer assignments with assignments to the Boolean variables of $(\star)$. Given a literal L, i.e., X_v or $\neg X_v$, we let $\texttt{lsum}(L)$ be the sum of all $x_{i,j}$ where $L_i^j \equiv L$ (we use 0 and 1 for *false* and *true*).

Lemma 6. ***(A)*** *If σ is a satisfying assignment for ϕ, then there is a non-conflicting assignment for $\mathcal{F}$ such that (1) $x_{i,1}+x_{i,2}+x_{i,3} = 1$ for all $1 \leq i \leq m$; (2) $\sigma(X_v) = 1 \Rightarrow \texttt{lsum}(\neg X_v) = 0$; and (3) $\sigma(X_v) = 0 \Rightarrow \texttt{lsum}(X_v) = 0$.* ***(B)*** *If ϕ is unsatisfiable, then for any non-conflicting assignment for $\mathcal{F}$ there is at least one $1 \leq i \leq m$ such that $x_{i,1} + x_{i,2} + x_{i,3} = 0$.*

Proof. **(A)** If σ satisfies ϕ, we construct a satisfying assignment for $\mathcal{F}$: first every $x_{i,j}$ is assigned the value of L_i^j, and then we turn some $x_{i,j}$ from 1 to 0 so that at most one variable of each clause is set to 1. Since we only turn 1s to 0s, when $\sigma(X_v) = 1$ (resp. $\sigma(X_v) = 0$) all constraint variables that correspond to $\neg X_v$ (resp. X_v) have value 0, and thus $\texttt{lsum}(\neg X_v) = 0$ (resp. $\texttt{lsum}(X_v) = 0$). **(B)** If $\mathcal{F}$ has a non-conflicting assignment in which $x_{i,1} + x_{i,2} + x_{i,3} = 1$ for all $1 \leq i \leq m$, then we can construct a satisfying assignment σ for ϕ in which $\sigma(X_v)$ is $\max\big(\{x_{i,j} | L_j^i \equiv X_v\} \cup \{1 - x_{i,j} | L_j^i \equiv \neg X_v\}\big)$, so ϕ is satisfiable. □

Next we proceed with the reduction, but first we give an outline. We build an integer loop, call it $\mathcal{T}$, with $2n+2$ abstract transitions: $2n$ transitions named $\Psi_{v,a}$, for $1 \leq v \leq n$ and $a \in \{0,1\}$; plus two named Φ and Ω. These are defined so that existence of a BMS-LLRF $\langle f_1, f_2 \rangle$ for $\mathcal{T}$ implies: (1) $\Psi_{v,0}$ and $\Psi_{v,1}$, for each $1 \leq v \leq n$, cannot be ranked by the same f_i, and the order in which they are ranked will represent a value for the existentially-quantified variable X_v; (2) Φ cannot be ranked by f_1, and it is ranked by f_2 iff $\forall X_{n+1} \ldots X_{2n}\ \neg\phi(X_1, \ldots, X_{2n})$ is true assuming the values induced for $X_1, \ldots, X_n$ in the previous step; and (3) Ω is necessarily ranked by f_1, its only role is to force Φ to be ranked by f_2. All these points will imply that $(\star)$ is true. For the other direction, if $(\star)$ is true we show how to construct a BMS-LLRF $\langle f_1, f_2 \rangle$ for $\mathcal{T}$. Next we formally define the variables and abstract transitions of $\mathcal{T}$, and prove the above claims.

Variables: Loop $\mathcal{T}$ includes $4m+2n+1$ variables: (1) every literal L_i^j contributes a variable $x_{i,j}$; (2) for each $1 \leq i \leq m$, we add a control variable $x_{i,0}$ which is used to check if clause C_i is satisfied; (3) for each $1 \leq v \leq n$, we add variables $z_{v,0}$ and $z_{v,1}$ which help in implementing the existential quantification; and (4) variable w, which helps in ranking the auxiliary transition Ω.

Transitions: First we define Φ, the transition that intuitively checks for satisfiability of $\phi(X_1, \ldots, X_{2n})$. It is a conjunction of the following constraints

$$0 \leq x_{i,j} \leq 1 \ \wedge \ x'_{i,j} = x_{i,j} \qquad \text{for all } 1 \leq i \leq m,\ 1 \leq j \leq 3 \tag{12}$$
$$x_{i,j} + x_{r,s} \leq 1 \qquad \text{for all } ((i,j),(r,s)) \in C \tag{13}$$
$$x_{i,0} \geq 0 \ \wedge \ x'_{i,0} = x_{i,0} + x_{i,1} + x_{i,2} + x_{i,3} - 1 \qquad \text{for all } 1 \leq i \leq m \tag{14}$$
$$z_{v,0} \geq 0 \ \wedge \ z'_{v,0} = z_{v,0} - \texttt{lsum}(X_v) \qquad \text{for all } 1 \leq v \leq n \tag{15}$$
$$z_{v,1} \geq 0 \ \wedge \ z'_{v,1} = z_{v,1} - \texttt{lsum}(\neg X_v) \qquad \text{for all } 1 \leq v \leq n \tag{16}$$
$$w' = w \tag{17}$$

Secondly, we define $2n$ transitions which, intuitively, force a choice of a Boolean value for each of $X_1, \ldots, X_n$. For $1 \leq v \leq n$ and $a \in \{0,1\}$, transition $\Psi_{v,a}$ is defined as a conjunction of the following constraints

$$z_{v,a} \geq 0 \ \wedge \ z'_{v,a} = z_{v,a} - 1 \tag{18}$$
$$z_{u,b} \geq 0 \qquad \text{for all } 1 \leq u \leq n, b \in \{0,1\},\ u \neq v \tag{19}$$
$$z'_{u,b} = z_{u,b} \qquad \text{for all } 1 \leq u \leq n,\ b \in \{0,1\},\ (u,b) \neq (v,a) \tag{20}$$
$$x'_{i,0} \geq 0 \ \wedge \ x'_{i,0} = x_{i,0} \qquad \text{for all } 1 \leq i \leq m \tag{21}$$
$$w \geq 0 \ \wedge \ w' = w \tag{22}$$

Finally we define the abstract transition Ω, which aids in forcing a desired form of the BMS-LLRF, and it is defined as a conjunction of the following constraints

$$w \geq 0 \ \wedge \ w' = w - 1 \tag{23}$$
$$z_{u,b} \geq 0 \ \wedge \ z'_{u,b} = z_{u,b} \qquad \text{for all } 1 \leq u \leq n,\ b \in \{0,1\} \tag{24}$$

Now, we argue that in order to have a two-component BMS-LLRF for $\mathcal{T}$, the transitions have to be associated to the two components in a particular way.

Lemma 7. *Suppose that $\langle f_1, f_2 \rangle$ is a* BMS-LLRF *for $\mathcal{T}$. Then, necessarily, the correspondence between the* BMS-LLRF *components and transitions is as follows: (i) Ω is ranked by f_1; (ii) Φ is ranked by f_2; (iii) for $1 \leq v \leq n$, one of $\Psi_{v,0}$ and $\Psi_{v,1}$ is ranked by f_1, and the other by f_2.*

Proof. An LRF for Ω must involve w, since it is the only decreasing variable, and cannot involve any $x_{i,j}$ since they change randomly. Similarly, an LRF for Φ cannot involve w as it has no lower bound, and it must involve at least one $x_{i,j}$ since no function that involves only $z_{v,a}$ variable(s) decreases for an initial state in which all $x_{i,j}$ are assigned 0. Note that such LRF cannot be non-increasing for Ω since $x_{i,j}$ change randomly in Ω. Thus, we conclude that Ω must be associated with f_1 and Φ with f_2. For the last point, for each $1 \leq v \leq n$, transitions $\Psi_{v,0}$ and $\Psi_{v,1}$ must correspond to different positions because variables that descend in one (namely $z_{v,a}$ of $\Psi_{v,a}$) are not bounded in the other (since (19) requires $u \neq v$). □

Lemma 8. *A* BMS-LLRF *of dimension two exists for $\mathcal{T}$ iff ($\star$) is true.*

Proof. Assume that a BMS-LLRF $\langle f_1, f_2 \rangle$ exists for $\mathcal{T}$, we show that ($\star$) is true. By Lemma 7 we know how the transitions are associated with the positions, up to the choice of placing $\Psi_{v,0}$ and $\Psi_{v,1}$, for each $1 \leq v \leq n$. Suppose that, for each $1 \leq v \leq n$, the one which is associated with f_2 is Ψ_{v,a_v}, i.e., $a_v \in \{0,1\}$, and let $\bar{a}_v$ be the complement of a_v. By construction we know that: (i) in Ψ_{v,a_v} the variables $z_{v,\bar{a}_v}$ and $x_{i,j}$ with $j \geq 1$ change randomly, which means that f_2 cannot involve them; and (ii) in Φ the variable w is not lower bounded, which means that f_2 cannot involve w. Since these transitions must be ranked by f_2, we can

assume that f_2 has the form $f_2(\mathbf{x},\mathbf{z},w) = \sum_i c_i \cdot x_{i,0} + \sum_v c_v \cdot z_{v,a_v}$ where c_i and c_v are non-negative rational coefficients. We claim that $(\star)$ is necessarily true; for that purpose we select the value a_v for each X_v, and next we show that this makes it is impossible to satisfy $\phi(X_1,\ldots,X_{2n})$. Assume, to the contrary, that there is a satisfying assignment σ for ϕ, such that $\sigma(X_v) = a_v$ for all $1 \leq v \leq n$. By Lemma 6 we know that we can construct an assignment to the variables $x_{i,j}$ such that (i) $x_{i,1} + x_{i,2} + x_{i,3} = 1$, for each $1 \leq i \leq m$, which means that $x'_{i,0} = x_{i,0}$ at (14); and (ii) for each $1 \leq v \leq m$, if $a_v = 0$ (resp. $a_v = 1$), then $\texttt{lsum}(X_v) = 0$ (resp. $\texttt{lsum}(\neg X_v) = 0$), which means that $z'_{v,a_v} = z_{v,a_v}$ at (15) (resp. (16)). Hence f_2 as described above does *not* rank Φ since none of its variables change, contradicting our assumption. We conclude that $(\star)$ is true.

Now assume that $(\star)$ is true, we construct a BMS-LLRF of dimension two. The assumption means that there are values $a_1,\ldots,a_n$ for the existentially-quantified variables to satisfy the sentence. Let $f_1(\mathbf{x},\mathbf{z},w) = w + \Sigma_{v=1}^{n} z_{v,\bar{a}_v}$ and $f_2(\mathbf{x},\mathbf{z},w) = \Sigma_{i=1}^{m} x_{i,0} + \sum_v z_{v,a_v}$. We claim that $\langle f_1, f_2\rangle$ is a BMS-LLRF such that: (i) f_1 is an LRF for Ω and $\Psi_{v,\bar{a}_v}$, and non-increasing for Ψ_{v,a_v} and Φ; and (ii) f_2 is an LRF for Ψ_{v,a_v} and Φ. All this is easy to verify, except possibly that f_2 is an LRF for Φ, for which we argue in more detail. By assumption, $\phi(a_1,\ldots,a_n,X_{n+1},\ldots,X_{2n})$ is unsatisfiable. Consider a state in which Φ is enabled; by (12,13), this state may be interpreted as a selection of non-conflicting literals. If one of the selected literals does not agree with the assignment chosen for $X_1,\ldots,X_n$, then by (15,16) the corresponding variable z_{v,a_v} is decreasing. Otherwise, there must be an unsatisfied clause, and the corresponding variable $x_{i,0}$ is decreasing. All other variables involved in f_2 are non-increasing, all are lower bounded, so f_2 is an LRF for Φ. □

Σ_2^P-hardness of BMS-LexLinRF$(d,\mathbb{Z})$ for $d = 2$ follows from Lemma 8. For $d > 2$, we add to $\mathcal{T}$ additional $d-2$ paths as those of Ex. 3; and to each original path in $\mathcal{T}$ we add $x'{=}x$ and $y'{=}y$ (x, y are used in Ex. 3). Then, the new loop has a BMS-LLRF of dimension d iff $(\star)$ is true. This concludes the proof of Th. 4.

7 Related Work

LLRFs appear in the classic works of Turing [28] and Floyd [15]. Automatic generation of LRFs and LLRFs for linear-constraint loops begins, in the context of logic programs, with Sohn and van Gelder [26]. For imperative programs, it begins with Colón and Sipma [9,10]. The work of Feautrier on scheduling [13,14] includes, in essence, generation of LRFs and LLRFs. All these works gave algorithms that yield polynomial time complexity (inherited from LP), except for Colón and Sipma's method which is based on LP duality and polars. The polynomial-time LP method later reappeared in [21,23]. These methods are complete over the rationals and can be used in an integer setting by relaxing the loop from integer to rational variables, sacrificing completeness. This completeness problem was pointed out (but not solved) in [21,24], while [11,13] pointed out the role of the integer hull in ensuring completeness. Bradley et al. [6] use a

bisection search over the space of coefficients for inferring LRFs over the integers, which yields completeness at exponential cost (as argued in [3]).

Alias et al. [1] extended the LP approach to LLRFs, obtaining a polynomial-time algorithm which is sound and complete over the rationals (for their notion of LLRF). The (earlier) work of Bradley et al. [4] introduced BMS-LLRFs and used a "constraint-solving method" that finds such LLRFs along with supporting invariants. The method involves an exponential search for the association of paths to LLRF components, and is complete over the reals. Subsequent work used more complex extensions of the LLRF concept [5,7]. Harris et al. [16] demonstrate that it is advantageous, to a tool that is based on a CEGAR loop, to search for LLRFs instead of LRFs only. The LLRFs they use are BMS-LLRFs. Similar observations have been reported in [12] (also using BMS-LLRFs), [8] (using ADFG-LLRFs) and [19] (using a an iterative construction that extends BMS-LLRFs). Heizmann and Leike [17] generalize the constraint-based approach by defining the concept of a "template" for which one can solve using a constraint solver. They also provide a template for ADFG-LLRFs (of constant dimension). Ben-Amram [2] shows that *every* terminating monotonicity-constraint program has a *piecewise* LLRF of dimension at most $2n$. Piecewise LLRFs are also used in [29], with no completeness result, there they are inferred by abstract interpretation.

8 Conclusion

This work contributes to understanding the design space of the ranking-function method, a well-known method for termination analysis of numeric loops, as well as related analyses (iteration bounds, parallelization schedules). This design space is inhabited by several kinds of "ranking functions" previously proposed. We focused on BMS-LLRFs and compared them to other proposals of a similar nature. We characterized the complexity of finding, or deciding the existence of, BMS-LLRF for rational and integer MLC loops. We also compared these three methods regarding the dimension of the LLRF, and the complexity of optimizing the dimension, which turns out to be essentially harder for BMS-LLRFs. Given our reductions, it is easy to show that it is impossible to approximate the minimal dimension of BMS-LLRFs, in polynomial time, within a factor *smaller than* $\frac{3}{2}$, unless $P{=}NP$ for rational loops, and $\Sigma_2^P{=}\Delta_2^P$ for integer loops (see App. C).

We conclude that none of the three methods is universally preferable. Even ADFG-LLRFs, which in principle are weaker than both other methods, have an advantage, in that the algorithm for computing them may be more efficient in practice (due to solving smaller LP problems). If this is not a concern, they can be replaced by BG-LLRFs, so we are left with two, incomparable techniques. This incomparability stems from the fact that BG-LLRFs and BMS-LLRFs relax the restrictions of ADFG-LLRFs in two orthogonal directions: the first in quantifying over concrete transitions rather than abstract ones, and the second in allowing negative components. By making both relaxations, we get a new type of LLRF [19], which is as in Def. 2 but relaxing condition (5) to hold only for $j = i$, but for which the computational complexity questions are still open.

References

1. Christophe Alias, Alain Darte, Paul Feautrier, and Laure Gonnord. Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs. In Radhia Cousot and Matthieu Martel, editors, *Static Analysis Symposium, SAS'10*, volume 6337 of *LNCS*, pages 117–133. Springer, 2010.
2. Amir M. Ben-Amram. Monotonicity constraints for termination in the integer domain. *Logical Methods in Computer Science*, 7(3), 2011.
3. Amir M. Ben-Amram and Samir Genaim. Ranking functions for linear-constraint loops. *Journal of the ACM*, 61(4), 2014.
4. Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. Linear ranking with reachability. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification, CAV'05*, volume 3576 of *LNCS*, pages 491–504. Springer, 2005.
5. Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. The polyranking principle. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *International Colloquium on Automata, Languages and Programming, ICALP'05*, volume 3580 of *LNCS*, pages 1349–1361. Springer, 2005.
6. Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. Termination analysis of integer linear loops. In Martín Abadi and Luca de Alfaro, editors, *Concurrency Theory, CONCUR 2005*, volume 3653 of *LNCS*, pages 488–502. Springer, 2005.
7. Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. Termination of polynomial programs. In Radhia Cousot, editor, *Verification, Model Checking, and Abstract Interpretation, VMCAI'05*, volume 3385 of *LNCS*, pages 113–129. Springer, 2005.
8. Marc Brockschmidt, Byron Cook, and Carsten Fuhs. Better termination proving through cooperation. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification, CAV 2013*, volume 8044 of *Lecture Notes in Computer Science*, pages 413–429. Springer, 2013.
9. Michael Colón and Henny Sipma. Synthesis of linear ranking functions. In Tiziana Margaria and Wang Yi, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'01*, volume 2031 of *LNCS*, pages 67–81. Springer, 2001.
10. Michael Colón and Henny Sipma. Practical methods for proving program termination. In Ed Brinksma and Kim Guldstrand Larsen, editors, *Computer Aided Verification, 14th International Conference, CAV'02,Copenhagen, Denmark, July 27-31, 2002, Proceedings*, volume 2404 of *LNCS*, pages 442–454. Springer, 2002.
11. Byron Cook, Daniel Kroening, Philipp Rümmer, and Christoph M. Wintersteiger. Ranking function synthesis for bit-vector relations. *Formal Methods in System Design*, 43(1):93–120, 2013.
12. Byron Cook, Abigail See, and Florian Zuleger. Ramsey vs. lexicographic termination proving. In Nir Piterman and Scott A. Smolka, editors, *Tools and Algorithms for the Construction and Analysis of Systems,TACAS 2013*, volume 7795 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2013.
13. Paul Feautrier. Some efficient solutions to the affine scheduling problem. I. one-dimensional time. *International Journal of Parallel Programming*, 21(5):313–347, 1992.
14. Paul Feautrier. Some efficient solutions to the affine scheduling problem. II. multi-dimensional time. *International Journal of Parallel Programming*, 21(6):389–420, 1992.
15. R. W. Floyd. Assigning meanings to programs. *Proceedings of Symposia in Applied Mathematics*, XIX:19–32, 1967.

16. William R Harris, Akash Lal, Aditya V Nori, and Sriram K Rajamani. Alternation for termination. In *Static Analysis Symposium, SAS 2011*, volume 6337 of *LNCS*, pages 304–319. Springer, 2011.
17. Matthias Heizmann and Jan Leike. Ranking templates for linear loops. In Erika Ábrahám and Klaus Havelund, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2014*, volume 8413 of *Lecture Notes in Computer Science*, pages 172–186. Springer International Publishing, 2014.
18. Michael Krivelevich and Benny Sudakov. Approximate coloring of uniform hypergraphs. *J. Algorithms*, 49(1):2–12, October 2003.
19. Daniel Larraz, Albert Oliveras, Enric Rodríguez-Carbonell, and Albert Rubio. Proving termination of imperative programs using Max-SMT. In *Formal Methods in Computer-Aided Design, FMCAD 2013*, pages 218–225. IEEE, 2013.
20. László Lovász. Coverings and colorings of hypergraphs. In *Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.
21. Frédéric Mesnard and Alexander Serebrenik. Recurrence with affine level mappings is P-time decidable for CLP(R). *TPLP*, 8(1):111–119, 2008.
22. Kevin T. Phelps and Vojtech Rödl. On the algorithmic complexity of coloring simple hypergraphs and steiner triple systems. *Combinatorica*, 4(1):79–88, 1984.
23. Andreas Podelski and Andrey Rybalchenko. A complete method for the synthesis of linear ranking functions. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation, VMCAI'04*, volume 2937 of *LNCS*, pages 239–251. Springer, 2004.
24. Andrey Rybalchenko. *Temporal Verification with Transition Invariants*. PhD thesis, Universität des Saarlandes, 2004.
25. Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, New York, 1986.
26. Kirack Sohn and Allen Van Gelder. Termination detection in logic programs using argument sizes. In Daniel J. Rosenkrantz, editor, *Symposium on Principles of Database Systems*, pages 216–226. ACM Press, 1991.
27. Larry J Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.
28. Alan M. Turing. Checking a large routine. In *Report of a Conference on High Speed Automatic Calculating Machines*, pages 67–69, 1948. reprinted in: The early British computer conferences, vol. 14 of Charles Babbage Institute Reprint Series For The History Of Computing, MIT Press, 1989.
29. Caterina Urban and Antoine Miné. An abstract domain to infer ordinal-valued ranking functions. In Zhong Shao, editor, *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*, pages 412–431. Springer, 2014.

A BMS-LexLinRF($\mathbb{Z}$) is coNP-complete

The coNP-hardness follows from the reduction in [3, Sect. 3.1], since it constructs a loop that either does not terminate or has an LRF. Next we prove inclusion in coNP by showing that the complement problem, i.e., the nonexistence of a BMS-LLRF, has a polynomially checkable witness. We assume a given MLC loop $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ where each $\mathcal{Q}_i$ is given as a set of linear constraints, over $2n$ variables (n variables and n primed variables). In this appendix we assume familiarity with Section 2.1 of [3] (preliminaries on polyhedra).

Recall that Proposition 1, when applied to $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$, implies that $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ has no BMS-LLRF iff there is a subset of the transition polyhedra that has no BMS-QLRF. This suggests that this subset can be used as a witness for the nonexistence of a BMS-LLRF. However, checking that such a subset has no BMS-QLRF cannot be done in polynomial time using the Algorithm of Lemma 4, since it requires computing the corresponding integer hull, and thus cannot be directly used as a witness. Instead, we show that there is finite set of integers points, related to this subset of the transition polyhedra, that can witness the nonexistence of a BMS-QLRF, and, moreover, can be checked in polynomial time (by checking that some corresponding set of constraints has no solution, over the rationals). Without loss of generality, assume that the subset of the transition polyhedra that we are considering, for the nonexistence of BMS-QLRF, is $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_\ell)$ for some $\ell \leq k$.

We first show that there is a polynomially checkable witness for the nonexistence of a BMS-QLRF for $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_\ell)$ that ranks a specific $I(\mathcal{Q}_p)$ for $1 \leq p \leq \ell$ — we refer to such BMS-QLRF as BMS$-$QLRF(p). Then we use this witness to construct one for the non-existence of BMS-QLRF.

Definition 4. *Let $X = X_1 \cup \cdots \cup X_\ell$, $Y = Y_1 \cup \cdots \cup Y_\ell$ and $1 \leq p \leq \ell$, such that (a) $X_i \subseteq I(\mathcal{Q}_i)$; (b) $Y_i \subseteq I(\mathcal{R}_{\mathcal{Q}_i})$; (c) $Y_i \neq \emptyset \Rightarrow X_i \neq \emptyset$; and (d) $X_p \neq \emptyset$. We say that $\langle X, Y \rangle$ is a witness against the existence of a* BMS$-$QLRF(p) *for $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_\ell)$ if the following set of linear constraints has no solution*

$$\vec{\lambda} \cdot \mathbf{x} + \lambda_0 \geq 0 \quad \textit{for all } \mathbf{x}'' \in X_p \tag{25a}$$

$$\vec{\lambda} \cdot (\mathbf{x} - \mathbf{x}') \geq 1 \quad \textit{for all } \mathbf{x}'' \in X_p \tag{25b}$$

$$\vec{\lambda} \cdot (\mathbf{x} - \mathbf{x}') \geq 0 \quad \textit{for all } \mathbf{x}'' \in X_i \quad (i \neq p) \tag{25c}$$

$$\vec{\lambda} \cdot \mathbf{y} \geq 0 \quad \textit{for all } \mathbf{y}'' \in Y_p \tag{25d}$$

$$\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') \geq 0 \quad \textit{for all } \mathbf{y}'' \in Y_i \quad \forall i \tag{25e}$$

The variables in the above constraints are $\lambda_0, \vec{\lambda}$, and they are rational-valued.

Lemma 9. *Let $X = X_1 \cup \cdots \cup X_\ell$, $Y = Y_1 \cup \cdots \cup Y_\ell$ and $1 \leq p \leq \ell$ be as in Definition 4. Then $I(\mathcal{Q}_1), \cdots, I(\mathcal{Q}_\ell)$ has no* BMS$-$QLRF(p).

Proof. Assume the contrary, i.e., there is $(\lambda_0, \vec{\lambda}) \in \mathbb{Q}^{n+1}$ such that $\rho(\mathbf{x}) = \vec{\lambda} \cdot \mathbf{x} + \lambda_0$ is a BMS$-$QLRF(p) for $I(\mathcal{Q}_1), \cdots, I(\mathcal{Q}_\ell)$. By assumption, (25a)-(25e)

has no solution, hence, they are not satisfied by the specific $(\lambda_0, \vec{\lambda})$ that we have chosen above. But (25a)-(25c) are clearly satisfied because ρ is a BMS$-$QLRF(p), and thus one of (25d) or (25e) is not satisfied. We reason on these two cases separately.

Case 1: Suppose (25e) is not satisfied, for some $\mathbf{y}'' \in Y_p$. That is, $\vec{\lambda} \cdot \mathbf{y} < 0$. Choose $\mathbf{x}'' \in X_p$, and note that for any integer $a \geq 0$, the integer point $\mathbf{z}'' = \mathbf{x}'' + a \cdot \mathbf{y}''$ is a transition in $I(\mathcal{Q}_p)$, and $\mathbf{z}'' = \binom{\mathbf{x}+a\cdot\mathbf{y}}{\mathbf{x}'+a\cdot\mathbf{y}'}$. Now,

$$\rho(\mathbf{z}) = \vec{\lambda} \cdot (\mathbf{x} + a \cdot \mathbf{y}) + \lambda_0 = \rho(\mathbf{x}) + a \cdot (\vec{\lambda} \cdot \mathbf{y})$$

It is easy to see that for sufficiently large a we get $\rho(\mathbf{z}) < 0$, since $\vec{\lambda} \cdot \mathbf{y} < 0$, which contradicts that ρ is BMS$-$QLRF(p).

Case 2: Suppose (25d) is not satisfied, for some $\mathbf{y}'' \in Y_i$. That is, $\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') < 0$. Choose $\mathbf{x}'' \in X_i$ and define $\mathbf{z}''$ as above. Now,

$$\rho(\mathbf{z}) - \rho(\mathbf{z}') = \vec{\lambda} \cdot ((\mathbf{x} + a \cdot \mathbf{y}) - (\mathbf{x}' + a \cdot \mathbf{y}')) = \rho(\mathbf{x}) - \rho(\mathbf{x}') + a \cdot (\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}'))$$

It is easy to see that for sufficiently large integer a we get $\rho(\mathbf{z}) - \rho(\mathbf{z}') < 0$, since $\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') < 0$, which contradicts that ρ is BMS$-$QLRF(p). This concludes the proof. □

Lemma 10. *If there no* BMS$-$QLRF(p) *for* $I(\mathcal{Q}_1), \cdots, I(\mathcal{Q}_\ell)$*, then there are finite sets* $X = X_1 \cup \cdots \cup X_\ell$ *and* $Y = Y_1 \cup \cdots \cup Y_\ell$*, fulfilling the conditions of Definition 4.*

Proof. For $1 \leq i \leq \ell$, let $\mathcal{Q}_{iI} = \text{convhull}\{X_i\} + \text{cone}\{Y_i\}$ be the generator representation of the integer hull of $\mathcal{Q}_i$, and define $X = X_1 \cup \cdots \cup X_\ell$ and $Y = Y_1 \cup \cdots \cup Y_\ell$. We claim that $\langle X, Y \rangle$, fulfill the conditions of Definition 4. Assume the contrary, i.e., (25a)-(25e) has a solution $(\lambda_0, \vec{\lambda}) \in \mathbb{Q}^{n+1}$, we show that $\rho(\mathbf{x}) = \vec{\lambda} \cdot \mathbf{x} + \lambda_0$ is a BMS$-$QLRF(p), contradicting the assumption that no BMS$-$QLRF(p) exists.

Pick a point $\mathbf{x}'' \in I(\mathcal{Q}_i)$, and let $X_i = \{\mathbf{x}''_1, \ldots, \mathbf{x}''_m\}$ and $Y_i = \{\mathbf{y}''_1, \ldots, \mathbf{y}''_t\}$. Note that $\mathbf{x}'' = \sum_{i=1}^{m} a_i \cdot \mathbf{x}''_i + \sum_{j=1}^{t} b_j \cdot \mathbf{y}''_j$ for some rationals $a_i, b_j \geq 0$, where $\sum_{i=1}^{m} a_i = 1$. We show that ρ correctly ranks $\mathbf{x}''$, i.e., fulfills the corresponding conditions of BMS-QLRF depending on if $\mathbf{x}''$ comes from $I(\mathcal{Q}_p)$ or from $I(\mathcal{Q}_i)$ with $i \neq p$:

- If $\mathbf{x}'' \in I(\mathcal{Q}_p)$, then each $x''_j \in X_i$ satisfies (25a,25b) and each $y''_j \in Y_i$ satisfies (25d,25e), then, it is easy to check that this necessarily imply $\rho(\mathbf{x}) \geq 0$ and $\rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 1$.
- If $\mathbf{x}'' \notin \mathcal{Q}_{pI}$, then each $x''_j \in X_i$ satisfies (25c) and each $y''_j \in Y_i$ satisfies (25e), it is easy to check that this necessarily imply $\rho(\mathbf{x}) - \rho(\mathbf{x}') \geq 0$.

This concludes the proof. □

Lemma 11. *If there is a finite witness for the nonexistence of* BMS−QLRF(p) *for* $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_\ell)$*, then there is one whose bit-size is polynomial in the bit-size of* $\mathcal{Q}_1, \ldots, \mathcal{Q}_\ell$*.*

Proof. By Lemma 10, we conclude that if there is a witness then there is one, $X = X_1 \cup \cdots \cup X_\ell$ and $Y = Y_1 \cup \cdots \cup Y_\ell$, such that X_i and Y_i come from the generator representation of $\mathcal{Q}_{iI}$.

Recall that (25a)-(25e) has no solution for the points of X and Y. A corollary of Farkas' Lemma [25, p. 94] states that if a finite set of inequalities over $\mathbb{Q}^d$, for some $d > 0$, has no solution, there is a subset of at most $d + 1$ inequalities that has no solution. Since the set of inequalities (25a)-(25e) is over $\mathbb{Q}^{n+1}$, there is a subset of at most $n + 2$ inequalities that has no solution.

These inequalities correspond to $n + 2$ points out of the sets X_i, Y_i. Let $\hat{X}_i$ (respectively $\hat{Y}_i$) be the set of points that come from X_i (respectively Y_i). Since (25a)-(25e) has no solution for these sets, at least one of the points must come from a set $\hat{X}_p$ (otherwise $\mathbf{0}$ is a solution). But $n + 1$ other points might come from sets $\hat{Y}_i$. Since a witness must satisfy $\hat{Y}_i \neq \emptyset \Rightarrow \hat{X}_i \neq \emptyset$ and $\hat{X}_p \neq 0$, we may have to add $n + 1$ points to form a valid witness, for a total of $2n + 3$. The bit-size of this witness is polynomial in the input bit-size, because each point comes from the generator representation of some $\mathcal{Q}_{iI}$, and it is known that it is possible to choose a generator representation in which each vertex has a bit-size that is polynomial in the bit-size of $\mathcal{Q}_i$ (see [3, Th. 2.7 and Th. 2.8]). □

Checking that a given $X = X_1 \cup \cdots \cup X_\ell$ and $Y = Y_1 \cup \cdots \cup Y_\ell$ is a witness as in Definition 4 can be done in polynomial time as follows: First we verify that each $\mathbf{x}'' \in X_i$ is in $I(\mathcal{Q}_i)$, which can be done by verifying $A_i\mathbf{x}'' \leq \mathbf{c}_i$; and that each $\mathbf{y}'' \in Y_i$ is in $I(\mathcal{R}_{\mathcal{Q}_i})$, which can be done by verifying $A_i\mathbf{y} \leq \mathbf{0}$. This is done in polynomial time. Note that according to Lemma 9 it is not necessary to check that X_i and Y_i come from a particular generator representation. Then we check that (25a)-(25e) has no solution, which can be done in polynomial time since it is an LP problem over $\mathbb{Q}^{n+1}$.

Corollary 2. *There is a polynomially checkable witness for the nonexistence of a* BMS-QLRF *for* $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_\ell)$*.*

Proof. The witness consists of ℓ witnesses, $\langle X^1, Y^1\rangle, \ldots, \langle X^\ell, Y^\ell\rangle$, each as in Definition 4 for some $1 \leq p \leq \ell$. Thus, the i-th one witnesses against the existence of BMS−QLRF(i). Thus all together witness against the existence of BMS-QLRF. Its size is clearly polynomial in the the input-bit size, and it can be checked in polynomial time by checking each $\langle X^i, Y^i\rangle$ as described before. □

Theorem 5. BMS−LexLinRF($\mathbb{Z}$) ∈ coNP *for MLC loops.*

Proof. Straightforward, given Proposition 1 and Corollary 2. □

B Complexity of the bounded-dimension decision problem for ADFG-LLRF and BG-LLRF

We assume a given MLC loop $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ where each $\mathcal{Q}_i$ is given as a set of linear constraints over $2n$ variables (n variables and n primed variables). The different bounded-dimension decision problems are denoted, naturally, by BG-LexLinRF$(d, \mathbb{Q})$, BG-LexLinRF$(d, \mathbb{Z})$, ADFG-LexLinRF$(d, \mathbb{Q})$, and ADFG-LexLinRF$(d, \mathbb{Z})$. In this appendix we assume familiarity with sections 2.1 and 5 of [3].

Theorem 6. BG-LexLinRF$(d, \mathbb{Q})$ *and* ADFG-LexLinRF$(d, \mathbb{Q})$ *are in P.*

Proof. We solve the problem by synthesizing an optimal-dimension BG-LLRF or ADFG-LLRF, which in both cases is PTIME. Then, we simply answer positively if and only if we found a tuple of dimension at most d. □

Next we move to BG-LexLinRF$(d, \mathbb{Z})$ and ADFG-LexLinRF$(d, \mathbb{Z})$, and show that both are coNP-complete. In both cases coNP-hardness is straightforward, since for $d = 1$ it becomes the problem of deciding if there is an LRF, and the argument can easily be extended to larger d. The rest of this section is dedicated to the inclusion in coNP.

Theorem 7. BG-LexLinRF$(d, \mathbb{Z})$ *and* ADFG-LexLinRF$(d, \mathbb{Z})$ *are in coNP.*

We prove for BG-LexLinRF$(d, \mathbb{Z})$, and then comment on how the proof can be adapted to ADFG-LexLinRF$(d, \mathbb{Z})$ as well.

The main step of the proof is to describe the form of a witness *against* the existence of a d-component BG-LLRF. The technical details of the proofs can be worked out exactly as in the corresponding proofs in Appendix A of this article, or in [3, Sec. 5.2]. Thus, we only sketch them here.

Lemma 12. *Let*

$$T_d \subseteq T_{d-1} \subseteq \cdots \subseteq T_1 \subseteq I(\mathcal{Q}_1) \cup \cdots \cup I(\mathcal{Q}_k),$$

such that (*i*) *there is no* LRF *for* T_d*; and* (*ii*) *for each* $\ell = 1, \ldots, d-1$, ***every quasi*-LRF** *for* T_ℓ *does not decrease on any of the transitions* $T_{\ell+1}$*. Then* $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ *has no* BG-LLRF *of dimension (at most)* d*. Conversely, if there is no* BG-LLRF *of dimension at most* d*, there is a chain of sets as above.*

Proof. ($\Rightarrow$) Suppose in contradiction that $\tau = \langle \rho_1, \ldots, \rho_d \rangle$ is a BG-LLRF (note that we can always pad the tuple to dimension d if it is of a smaller dimension). Then ρ_1 is a quasi-LRF for T_1, and so by (ii) does not decrease on T_2. Hence $\langle \rho_2, \ldots, \rho_d \rangle$ is a BG-LLRF for T_2. Proceedings in this way we deduce that ρ_d must be an LRF for T_d, contradicting (i).

($\Leftarrow$) Suppose that there is no BG-LLRF of dimension at most d. Following the BG-LLRF (synthesis) algorithm [3, Alg. 1, p.30], we see that one of the following must happen: (1) within d recursive calls, the algorithm fails to find

a non-trivial quasi-LRF, or (2) a $d+1$ recursive call is reached. We construct sets $T_1, \ldots, T_d$ that satisfy (i,ii) as follows: Let $\langle \mathcal{P}_{j1}, \ldots, \mathcal{P}_{jk} \rangle$, for $1 \leq j \leq d$, be the parameters received by the BG-LLRF algorithm in j-th invocation (if the algorithm stops at iteration $s < d$, we assume $\mathcal{P}_{ji} = \mathcal{P}_{si}$ for any $s < j \leq d$); and define $T_j = \cup_{i=1}^{k} I(\mathcal{P}_{ji})$, for $1 \leq j \leq d$. □

In what follows, given sets of integer points $X' \subseteq X$ and $Y' \subseteq Y$, we let $\Gamma(X, Y, X', Y')$ be the conjunction of the following inequalities:

$$\vec{\lambda} \cdot \mathbf{x} + \lambda_0 \geq 0 \quad \text{for all } \mathbf{x}'' \in X \tag{26a}$$

$$\vec{\lambda} \cdot \mathbf{y} \geq 0 \quad \text{for all } \mathbf{y}'' \in Y \tag{26b}$$

$$\vec{\lambda} \cdot (\mathbf{x} - \mathbf{x}') \geq 0 \quad \text{for all } \mathbf{x}'' \in X \tag{26c}$$

$$\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') \geq 0 \quad \text{for all } \mathbf{y}'' \in Y \tag{26d}$$

$$\sum_{\mathbf{x}'' \in X'} \vec{\lambda} \cdot (\mathbf{x} - \mathbf{x}') + \sum_{\mathbf{y}'' \in Y'} \vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') \geq 1 \tag{26e}$$

Intuitively, $\langle X, Y \rangle$ and $\langle X', Y' \rangle$ will be generators of sets of integer points $T' \subseteq T$ such that the solutions of $\Gamma(X, Y, X', Y')$ are the quasi-LRFs of T that also decrease on some points of T'. For sets of integer points X and Y we let $\Psi(X, Y)$ be the conjunction of the following inequalities:

$$\vec{\lambda} \cdot \mathbf{x} + \lambda_0 \geq 0 \quad \text{for all } \mathbf{x}'' \in X \tag{27a}$$

$$\vec{\lambda} \cdot \mathbf{y} \geq 0 \quad \text{for all } \mathbf{y}'' \in Y \tag{27b}$$

$$\vec{\lambda} \cdot (\mathbf{x} - \mathbf{x}') \geq 1 \quad \text{for all } \mathbf{x}'' \in X \tag{27c}$$

$$\vec{\lambda} \cdot (\mathbf{y} - \mathbf{y}') \geq 0 \quad \text{for all } \mathbf{y}'' \in Y \tag{27d}$$

Intuitively, $\langle X, Y \rangle$ will generate a set of integer points T, and the solutions of $\Psi(X, Y)$ are all LRFs of T.

Definition 5. *Given $\langle X_1, Y_1 \rangle, \ldots, \langle X_d, Y_d \rangle$, where $X_j = X_{j1} \cup \cdots \cup X_{jk}$ and $Y_j = Y_{j1} \cup \cdots \cup Y_{jk}$, such that*

(a) $X_{di} \subseteq X_{(d-1)i} \subseteq \cdots \subseteq X_{1i} \subseteq I(\mathcal{Q}_i)$;
(b) $Y_{di} \subseteq Y_{(d-1)i} \subseteq \cdots \subseteq Y_{1i} \subseteq I(\mathcal{R}_{\mathcal{Q}_i})$;
(c) $Y_{ji} \neq \emptyset \Rightarrow X_{ji} \neq \emptyset$.

We say that $\langle X_1, Y_1 \rangle, \ldots, \langle X_d, Y_d \rangle$ form a witness against the existence of a BG-LLRF *of dimension at most d for $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ if it satisfies the following requirements:*

(d) $\Psi(X_d, Y_d)$ has no solution; and
(e) $\Gamma(X_i, Y_i, X_{i+1}, Y_{i+1})$, for any $1 \leq i \leq d-1$, has no solution.

Each $\langle X_j, Y_j \rangle$ corresponds to a set of integer points T_j such that $T_{j+1} \subseteq T_j$. In addition, condition (d) guarantees that T_d has no LRF, and condition (e) guarantees that there is no quasi-LRF for T_j that is decreasing for some points of T_{j+1}.

Lemma 13. *Let $\langle X_1, Y_1\rangle, \ldots, \langle X_d, Y_d\rangle$ be as in Definition 5. Then there are $T_d \subseteq \cdots \subseteq T_1 \subseteq I(\mathcal{Q}_1) \cup \cdots \cup I(\mathcal{Q}_k)$ satisfying the requirements of Lemma 12.*

Proof. We construct sets of transitions $T_d \subseteq T_{d-1} \subseteq \cdots \subseteq T_1 \subseteq I(\mathcal{Q}_1) \cup \cdots \cup I(\mathcal{Q}_k)$ that satisfy the requirements of Lemma 12. We construct T_j from $\langle X_j, Y_j\rangle$ as follows:

$$T_j = \{\mathbf{x}'' + a\mathbf{y}'' \mid \mathbf{x}'' \in X_{ji}, \mathbf{y}'' \in Y_{ji}, \text{ integer } a \geq 0 \} .$$

Note that for $\mathbf{x}'' \in X_{ji}$ and $\mathbf{y}'' \in Y_{ji}$, the point $\mathbf{x}'' + a\mathbf{y}''$, for any integer $a \geq 0$, is a transition in $I(\mathcal{Q}_i)$, thus $T_j \subseteq I(\mathcal{Q}_1) \cup \cdots \cup I(\mathcal{Q}_k)$. We claim that these sets satisfy the requirements of Lemma 12; the proof can be worked out similarly to [3, Lemma 5.18]). □

The last result states that our witnesses are sound—they really imply that there is no BG-LLRF of the desired dimension. Next we should also prove that when there is no such BG-LLRF, witness sets as above exist, and their size can be polynomially bounded.

Lemma 14. *Suppose that $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ has no* BG-LLRF *of dimension at most d. Then there are $\langle X_1, Y_1\rangle, \ldots, \langle X_d, Y_d\rangle$ of bit-size polynomially bounded by the bit-size of the input transition polyhedra (as constraints), fulfilling the conditions of Definition 5.*

Proof. Consider again the BG-LLRF algorithm [3, Alg. 1,p.30], if the integer loop $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$ has no BG-LLRF of dimension at most d, one of the following happens: (1) within d recursive calls, the algorithm fails to find a non-trivial quasi-LRF, or (2) a $d+1$ recursive call is reached. Let $\langle \mathcal{P}_{j1}, \ldots, \mathcal{P}_{jk}\rangle$, for $1 \leq j \leq d$, be the parameters received by the BG-LLRFs algorithm in j-th recursive call (if the algorithm stops at iteration $s < d$, we let $\mathcal{P}_{ji} = \mathcal{P}_{si}$ for any $s < j \leq d$). Define $T_j = \cup_{i=1}^{k} I(\mathcal{P}_{ji})$, for all $1 \leq j \leq d$. Then, clearly $T_1, \ldots, T_d$ are sets of transitions that satisfy the requirements of Lemma 12. We construct a witness that corresponds to these sets as follows: First note that each $\mathcal{P}_{ji}$ is integral, and has a corresponding generator representation

$$\mathcal{P}_{ji} = \text{convhull}\{X_{ji}\} + \text{cone}\{Y_{ji}\} .$$

where X_{ji} and Y_{ji} are finite sets of integer points. Then, we define each component $\langle X_j, Y_j\rangle$ of the witness as $X_j = X_{j1} \cup \ldots \cup X_{jk}$ and $Y_j = Y_{j1} \cup \ldots \cup Y_{jk}$.

This witness satisfies condition (c) of Definition 5, because we may assume that none of the transition polyhedra is a cone (otherwise the loop clearly does not terminate), and thus $X_{ij} \neq \emptyset$. To show that it satisfies conditions (a,b) as well, we rely on the following fact [25, p.107]: if a polyhedron $\mathcal{P} = \text{convhull}\{X\} + \text{cone}\{Y\}$ is a face of a polyhedron $\mathcal{P}' = \text{convhull}\{X'\} + \text{cone}\{Y'\}$, then $X \subseteq X'$ and $Y \subseteq Y'$. Now a property of the BG-LLRF algorithm [3, Lemma 5.8] is that $\mathcal{P}_{(j+1)i}$ is a face of $\mathcal{P}_{ji}$, and thus $X_{(j+1)i} \subseteq X_{ji}$ and $Y_{(j+1)i} \subseteq Y_{ji}$, so the witness satisfies conditions (a,b). Showing that conditions (d,e) hold can be worked out as in [3, Lemma 5.19]. Finally, We can reduce the witness above to polynomial bit-size, using the same arguments as [3, Lemma 5.22]. □

Checking a witness can be done in polynomial time as follows: First we verify that each $\mathbf{x}'' \in X_{ji}$ is in $I(\mathcal{Q}_i)$, which can be done by verifying $A_i\mathbf{x}'' \leq \mathbf{c}_i$; and that each $\mathbf{y}'' \in Y_{ji}$ is in $I(\mathcal{R}_{\mathcal{Q}_i})$, which can be done by verifying $A_i\mathbf{y}'' \leq \mathbf{0}$. This is done in polynomial time. Checking that $\Psi(X_d, Y_d)$ and $\Gamma(X_i, Y_i, X_{i+1}, Y_{i+1})$ have no solution can be done in polynomial time since it is an LP problem over the rationals. This concludes the proof of Theorem 7, and thus BG-LexLinRF$(d, \mathbb{Z})$ is coNP-complete.

B.1 The case of ADFG-LLRFs

Next we explain how to adapt the above proof to ADFG-LLRFs. The same approach works for adapting the coNP-completeness proof of [3] for the existence of BG-LLRFs to the case of ADFG-LLRFs.

The important difference between the quasi-LRFs used in ADFG-LLRF from those of BG-LLRFs is that they the must be non-negative over all $I(\mathcal{Q}_1), \ldots, I(\mathcal{Q}_k)$. This means that when checking that a witness has no solution (signifying that there is no LRF, or no quasi-LRF with a certain non-triviality restriction) we should take this additional restriction into account. This can be done by extending the witness with an extra component $X' = X'_1 \cup \cdots \cup X'_k$ and $Y' = Y'_1 \cup \cdots \cup Y'_k$, such that (*i*) $X'_i \subseteq I(\mathcal{Q}_i)$ and finite; (*ii*) $Y'_i \subseteq I(\mathcal{R}_{\mathcal{Q}_i})$ and finite; and (*iii*) $Y'_i \neq \emptyset \Rightarrow X'_i \neq \emptyset$. In addition, we add the the following in inequalities requirements from a witness

$$\vec{\lambda}\cdot\mathbf{x} + \lambda_0 \geq 0 \quad \text{for all } \mathbf{x}'' \in X' \tag{28a}$$

$$\vec{\lambda}\cdot\mathbf{y} \geq 0 \quad \text{for all } \mathbf{y}'' \in Y' \,. \tag{28b}$$

C Approximation of Minimum Dimension

Since finding out whether a BMS-LLRF of dimension d exists is NP-hard and Σ_2^P-hard, for rational and integer loops, respectively, a natural question to ask is if we can approximate the minimum dimension in polynomial time.

Rational loops. It is know that it is impossible to approximate, in polynomial time, the chromatic number of r-uniform hypergraphs on n vertices within a factor $n^{1-\varepsilon}$, for any $\varepsilon > 0$, unless $NP \subseteq ZPP$ [18]. Given our reduction, we conclude that it is impossible to approximate the minimal dimension of BMS-LLRFs within a factor $k^{1-\varepsilon}$, for any $\varepsilon > 0$, unless $NP \subseteq ZPP$ (recall that n vertices generate k paths in our reduction). Similarly, we cannot do such approximation within a factor *smaller than* $\frac{3}{2}$, unless $P = NP$, because we can then decide if a 3-uniform hypergraph has 2-coloring, which is an NP-complete [20].

Integer loops. A polynomial algorithm, even given access to an NP oracle (a SAT solver) for free, cannot approximate the minimum dimension d of BMS-LLRFs within a factor *smaller than* $\frac{3}{2}$, unless $\Sigma_2^P = \Delta_2^P$. This is because if such an algorithm exists, then for the loop $\mathcal{T}$, a result of 2 will mean that $(\star)$ is true, and any other result (necessarily 3 or 4, since it has to be under $\frac{3}{2} \cdot 3$) will mean that it is false. Thus a Σ_2^P-hard problem is solved in Δ_2^P complexity.