DOI: 10.1007/s00165-015-0333-3

DOI 10.1007/s00165-015-0333-3
BCS © 2015
Formal Aspects of Computing (2015) 27: 805–829

Formal Aspects of Computing

Generating invariants for non-linear loops by linear algebraic methods

Rachid Rebiha[1], Arnaldo Vieira Moura[1] and Nadir Matringe[2]

[1] Institute of Computing, University of Campinas, Campinas, Brazil
[2] LMA, University of Poitiers, Poitiers, France

Abstract. We present new computational methods that can automate the discovery and the strengthening of non-linear interrelationships among the variables of programs containing non-linear loops, that is, that give rise to multivariate polynomial and fractional relationships. Our methods have complexities lower than the mathematical foundations of the previous approaches, which used Gröbner basis computations, quantifier eliminations or cylindrical algebraic decompositions. We show that the preconditions for discrete transitions can be viewed as morphisms over a vector space of degree bounded by polynomials. These morphisms can, thus, be suitably represented by matrices. We also introduce fractional and polynomial consecution, as more general forms for approximating consecution. The new relaxed consecution conditions are also encoded as morphisms represented by matrices. By so doing, we can reduce the non-linear loop invariant generation problem to the computation of eigenspaces of specific morphisms. Moreover, as one of the main results, we provide very general sufficient conditions allowing for the existence and computation of whole loop invariant ideals. As far as it is our knowledge, it is the first invariant generation methods that can handle multivariate fractional loops.

Keywords: Formal methods, Invariant generation, Linear algebra

1. Introduction

An invariant at a location of a program is an assertion true of any reachable program state associated to this location. We present a new method for non-linear invariant generation that addresses various deficiencies of other state-of-the-art methods. More generally, we provide mathematical techniques and design efficient algorithms to automate the discovery and strengthening of non-linear interrelationships among the variables of programs containing non-linear loops, which lead to multivariate polynomial and fractional relationships.

FAPESP BEPE grant 2013/04734-9, FAPESP grant 2011/08947-1.
Correspondence and offprint requests to: R. Rebiha, E-mail: rachid@ic.unicamp.br

It is well-known that the automation and effectiveness of formal verification depend on the ease with which invariants can be automatically generated. Actually, the verification problem of safety properties, such as no null pointer deferenciation, buffer overflows, memory leak or outbounds, and array accesses, can be reduced to the problem of invariant generation [MP95]. Invariants are also essential to prove and establish liveness properties such as progress or termination [MP95]. Furthermore, the standard techniques [MP95] for program verification use invariant assertions to directly prove program properties, or to provide supporting lemmas that can be used to establish other safety and liveness properties. We look for invariants that strengthen what we wish to prove, and so allow us to establish the desired property. Also, they can provide precise over-approximations to the set of reachable states. Also, the weakest precondition method [Dij76, Flo67], the Floyd–Hoare inductive assertion technique [Flo67, Hoa69], and the standard ranking functions technique [MP95], all require loop invariants in order to establish correctness and so render the verification method completely automatic. Again, in order to establish termination verification, the standard ranking functions technique requires the automatic generation of invariants.

In order to generate loop invariants, one need to discover *inductive* assertions that hold at any step of the loop. An inductive assertion also holds at the first time the loop location is reached—this is the initiation condition—and it is also preserved under instructions that cycle back to the loop location—this being the consecution condition. If we choose transition systems as the representation model and automata as the computational model, we can say that the invariant holds in the initial state of the system—the initial condition—and that every possible transition preserves it—the consecution conditions. In other words, the invariant holds in any possible reachable state.

In the case of loops describing a linear system, Farka's lemma [Sch86] can be used to encode the conditions for *linear* invariants. On the other hand, for *non-linear* invariants, the difficulty of automatic generation remains very challenging. By today known methods, they require a large number of Gröbner bases computations [SSM04b], first-order quantifier eliminations [Wei97, Col75], or cylindrical algebraic decompositions [CXYZ07]. Invariants can also be computed as fixed points on ideals, using fixed point techniques [RCK07a], abstract interpretation frameworks [CC92, CC77], and Gröbner bases constructions. Abstract interpretation introduces imprecision, and widening operators must be provided manually by the user in order to assure termination. A too coarse abstraction would limit these approaches to trivial invariants in the presence of non-linear loops. Other methods [KJ06, Kov08] attempt to generate invariants from a restricted class of P-solvable loops. These methods use techniques from algebra and combinatorics, like Gröbner bases [JKP06], variable elimination, algebraic dependencies and symbolic summation, and so also incur in high computational complexities.

More recent approaches have been constraint-based [SSM04b, RCK07a, Kap04, RCK07b, SSM04a, GT08, PJ04]. In these cases, a candidate invariant with a fixed degree and unknown parametric coefficients, i.e., a template form, is proposed as the target invariant to be generated. The conditions for invariance are then encoded, resulting in constraints on the unknown coefficients whose solutions yield invariants. One of the main advantage of such constraint-based approaches is that they are goal-oriented. The main challenge for these techniques remains in the fact that they still require a high number of Gröbner bases [Buc96] computations, first-order quantifier elimination [Wei97, Col75], cylindrical algebraic decomposition [CXYZ07], or abstraction operators. And known algorithms for those problems are, at least, of double exponential complexity.

Despite tremendous progress over the years [SSM04b, BBGL00, RCK07a, BLS96, CXYZ07, Kov08, KJ06, Cou05, MOS02, RCK07b, GT08, Tiw08, PC08], the problem of loop invariant generation remains very challenging for non-linear discrete systems. In this work we present new methods for the automatic generation of loop invariants for non-linear systems. As will be seen, these methods give rise to more efficient algorithms, with much lower complexity in space and time. We develop the new methods by first extending our previous work on non-linear non-trivial invariant generation for discrete programs with nested loops and conditional statements, [RMM08b, RMM10].

We can summarize our contributions as follows:

- We do not need to start with candidate invariants that generate intractable solving problems. Instead, we show that the preconditions for discrete transitions can be viewed as morphisms over a vector space of degree bounded by polynomials which can, thus, be suitably represented by matrices.
- We introduce more general forms for approximating consecution, called fraction and polynomial consecution. The new relaxed consecution requirements are also encoded as morphisms, represented by matrices with terms that are the unknown coefficients used to approximate the consecution conditions. As far as it is our knowledge, these are the first methods that can effectively handle multivariate fractional systems.

- We succeed in reducing the non-linear loop invariant generation problem to the computation of eigenspaces or nullspaces of specific endomorphisms. We provide general sufficient conditions guaranteeing the existence and allowing the computation of invariant ideals. The unknown coefficients appearing in the matrices used to approximate the consecution conditions are assigned in order to insure that the nullspaces generated are not trivial ones. Taking into consideration the specific type of matrices we are manipulating, we determine for which values of the coefficients their ranks are minimal. Our decision procedure for those assignments is very simple and efficient. At each step of the assignments, we echelon the matrices by making the highest term of one column to vanish.
- Our approach does not generate an invariant at a time. Instead we generate an ideal of invariants—an infinite structure—by computing the basis of a specific vector space giving rise to provable, inductive invariants. This could also be used by existing approaches dealing with the generations of such vector spaces of inductive invariants [Cou05, RCK07b, Kov08].
- Our technique comprises three computational steps, each of polynomial time complexity. In contrast, the most recent and best performing constraint-based approaches can be summarized in three main steps, with each of these steps inducing a number of computations that are of double exponential time complexity. Further, as soon as the loop contains non-linear instructions, the constraints considered at the final step gives rise to systems of non-linear equations, rendering unfeasible their resolution; see Sect. 4.3. We, by contrast, propose a computational method of much lower time complexity than other present approaches based on fixed point computation, or on constraint-based approaches.
- We present some preliminary experimental results. For that, we used `Sage` [SJ05] with interfaces written in `Python`, in order to be able to access other mathematical packages.
- We incorporate a strategy that attains optimal degree bounds for candidate invariants. We also note that our existence results and methods can be reused in other approaches in order to reduce their time complexity, since they can reduce the number of Gröbner basis computations or quantifier eliminations, for example.

Example 1 (*Motivational Example*). Consider the following program loop:

```
...
while (...){
...
  x := x*y + x;
  ...
  y := y^2;
...
}
```

Present constraint-based static program analysis techniques are facing some difficulties in producing any conclusion that could be somehow related to the values of the variables x and y, given that the semantic of the two instructions inside the loop relies on non-linear arithmetic. Such non-linearities are presently recognized by industry and academia as a critical bottleneck for automatic program verification and static program analysis.

In present standard approaches for invariant generation for non-linear loops, the loop instructions are first used in order to form varieties, to build associated algebraic assertions and an ideal I. Then, they compute a Gröbner basis G for I. Next, they postulate a template polynomial Q, i.e., a polynomial with unknown coefficients, as a *candidate invariant*, and proceed by performing a reduction of Q by G in order to obtain its reduced normal form $NF_G(Q)$. An important obstacle faced at this point is that all known algorithms for computing Gröbner basis and for constructing the normal form reduction $NF_G(Q)$ are of doubly exponential time complexity. Having the normal form $NF_G(Q)$, they generate the set of *candidate invariant constraints* in the form of a system of equations by letting $NF_G(Q) = 0$, and then they attempt to solve it directly. But we show that as soon as the loop contains a non-linear instruction, the constraints obtained in their final step lead to systems of non-linear equations in unknown parameters, which remains untractable in practice (see Sect. 4.3). For more details on the limitations of such techniques, illustrated on this same motivational example, see Example (6), in Sect. 4.3.

In this article, we introduce new symbolic techniques with fast numerical approaches that can be used in these situations. Our techniques have fewer computational steps. We first compute some specific matrix M obtained directly from the loop instructions. We then generate a matrix L that we use to approximate the consecution condition. Matrix L contains some fixed parameters so as to guarantee that such nullspaces are not empty. We note that the unknown coefficients do not play the role of templates. Rather, they are introduced to allow us to reduce the rank of the matrix $M-L$, thus leading to a non-trivial nullspace. Based on our theoretical contributions, we know that the nullspace of $M-L$ provides us with a non-trivial vector space of inductive invariants. In the example at hand, our method directly computes $\{x^2, x*y-x, y^2-2y+1\}$ as a basis for a vector space of invariants, and all elements in this space provide non-trivial invariants. We thus obtain an ideal for non-trivial inductive invariants. In other words, for all $G_1, G_2, G_3 \in \mathbb{R}[x, y]$, we would get $G_1(x, y)(x^2)+G_2(x, y)(xy-x)+G_3(x, y)(y^2-2y+1)=0$ as an inductive invariant. Take, for instance, the initial step $(y = y_0, x = 1)$. A possible invariant is, then, $y_0(1-y_0)x^2 + xy - x + y^2 - 2y + 1 = 0$. By taking two elements of this basis, one could generates inductive invariants holding for any type of initial conditions on the variables. Such invariants are beyond the reach of other current invariant generation techniques. In Sect. 5.4, we make explicit all the computational steps of our method. □

In Sect. 2 we present ideals of polynomials and their possible interactions with inductive assertions. In Sect. 3 we introduce new consecution conditions, and extend them to fractional systems. In Sect. 4 we consider linear loops, and present results for the existence of *non-trivial* invariants in these settings. We also recast the problem in term of linear algebra, and present a complete decision procedure for the automatic generation of *non-trivial non-linear* invariants. In Sect. 5 we extend our method to non-linear loops. In Sect. 6 we propose a strategy to obtain optimal degree bounds. In Sect. 7 we provide a complete generalization by considering loops describing multivariate fractional systems, and in Sect. 8 we show how to handle conditions and nested loops. Section 9 exposes some preliminary experimental results, and Sect. 10 contains a discussion. We conclude in Sect. 11. The Appendix contains proofs for all theorems, lemmas and corollaries stated in this article. Further examples can be found in companion technical reports and other articles [RMM08a, RMM08b, RMM10, RM11a, RM11b].

2. Polynomial ideals and inductive assertions

We will use the following notations. Let $\mathbb{K}$ be a field. The ring of multivariate polynomials over the set of variables $\{X_1, .., X_n\}$ with coefficients in $\mathbb{K}$ will be indicated by $\mathbb{K}[X_1, .., X_n]$. We will denote by $\mathbb{R}_d[X_1, .., X_n]$ the vector space of multivariate polynomials of degree at most d over the set of real variables $\{X_1, .., X_n\}$. We will write $Vect(v_1, \ldots, v_n)$ for the vector space generated by a basis $(v_1, \ldots, v_n)$. The dimension of a subspace $W \subseteq Vect(v_1, \ldots, v_n)$ is written $Dim(W)$. Clearly, $Dim(Vect(v_1, \ldots, v_n)) = n$. The vector space of all matrices over a filed $\mathbb{K}$ will be denoted by $\mathcal{M}(m, n, \mathbb{K})$. Let $M \in \mathcal{M}(m, n, \mathbb{K})$ be the matrix representation of a morphism over a vector space. Its *kernel*, or *nullspace*, is the set $Ker(M) = \{v \in \mathbb{K}^n \mid M \cdot v = 0_{\mathbb{K}^m}\}$. The kernel of M is said to be trivial if it contains only the zero vector. The rank of M, denoted $Rank(M)$, is the dimension of the subspace $\{M \cdot v \in \mathbb{K}^m \mid v \in \mathbb{K}^n\}$. Alternatively, it is the number of linearly independent columns or rows of the matrix. We know that $Rank(M) + Dim(Ker(M)) = n$. An *eigenvalue* of M is a scalar $\lambda \in \mathbb{K}$ such that $M \cdot v = \lambda\, v$ for some nonzero vector v. The set $\{v \in \mathbb{K}^n \mid M \cdot v = \lambda\, v\}$ is the *eigenspace* associated to an eigenvalue λ. To compute the basis of eigenspaces and nullspaces, we use well-known state-of-the-art algorithms, such as those that `Sage` or `Mathematica` provide. To solve equations of degree less than 5 one could consider the classical Lagrange resolvents [Lan02, AV97] method. A primed x' will refer to the next state value of a variable x, after a transition is taken. If V is a set of variables, then V' is the set of all primed variables in V.

2.1. Polynomial ideals

Definition 1 An ideal is any set $I \subseteq \mathbb{K}[X_1, .., X_n]$ such that

- It is closed under addition. In other words, if $P, Q \in I$ then $P + Q \in I$;
- It is closed under multiplication by any element in $\mathbb{K}[X_1, .., X_n]$, *i.e.*, if $P \in I$ and $Q \in K[X_1, .., X_n]$ then $PQ \in I$;
- It includes the null polynomial, *i.e.* $0_{\mathbb{K}[X_1,..,X_n]} \in I$.

□

Let $E \subseteq \mathbb{K}[X_1, .., X_n]$ be a set of polynomials. The *ideal generated by* E is the set of finite sums

$$(E) = \left\{ \sum_{i=1}^{k} P_i Q_i \;\mid\; P_i \in \mathbb{K}[X_1, \ldots, X_n],\;\; Q_i \in E,\;\; k \geq 1 \right\}.$$

Definition 2 A set of polynomials E is said to be a *basis* of an ideal I if $I = (E)$. □

By the Hilbert Basis Theorem, we know that all ideals have a *finite basis*. Let I be an ideal and Q a polynomial. The important question of knowing if Q belongs to I is known as the *Ideal Membership Problem*. In order to decide membership, one first computes the normal form of Q by performing polynomial reductions according to I. If the resulting normal form is the null polynomial we can conclude that $Q \in I$. A *Gröbner basis* of I guarantees the confluence and termination of those polynomial reductions.

2.2. Inductive assertions and invariants

The contribution and novelty in our approach clearly set it apart from [SSM04b] as their constraint-based techniques require several Gröbner basis computations and also require solving non-linear problems for each location. Nevertheless, they introduce a useful formalism to treat programs loops, and we start from similar definitions for transitions systems, inductive invariants and consecution conditions. We will use *transition systems* as representation of imperative programs and *automata* as their computational models.

Definition 3 A *transition system* is given by $\langle V, L, \mathcal{T}, l_0, \Theta \rangle$, where

- V is a set of variables,
- L is a set of locations and $l_0 \in L$ is the initial location.
- A *transition* $\tau \in \mathcal{T}$ is given by a tuple $\langle l_{pre}, l_{post}, \rho_\tau \rangle$, where l_{pre} and l_{post} name the pre- and post- locations of τ, and the transition relation ρ_τ is a first-order assertion over $V \cup V'$.
- Θ is the initial condition, given as a first-order assertion over V.

The transition system is *affine* when ρ_τ is an affine form, and it is *algebraic* when ρ_τ is an algebraic form. □

Definition 4 Let W be a transition system. An *invariant* at location $l \in L$ is an assertion over V which holds at all states reaching location l. An *invariant* of W is an assertion over V that holds at all locations. □

Given our representational and computational models, we want to say that an invariant holds in the initial state of the system, a condition that will be guaranteed by an initial condition. We also want to say that every possible transition preserves the invariant, when specific consecution conditions hold. In order to generate loop invariants one needs to discover *inductive* assertions.

Definition 5 Let $W = \langle V, L, \mathcal{T}, l_0, \Theta \rangle$ be a transition system and let $\mathbb{D}$ be an assertion domain. An assertion map for W is a map $\eta : L \to \mathbb{D}$. We say that η is *inductive* if and only if the following conditions hold:

- *Initiation*: $\Theta \models \eta(l_0)$
- *Consecution*: For all τ in $\mathcal{T}$ s.t. $\tau = \langle l_i, l_j, \rho_\tau \rangle$ we have $\eta(l_i) \wedge \rho_\tau \models \eta(l_j)'$.

□

We know that if η is an inductive assertion map then $\eta(l)$ is an invariant at l for W [Flo67].

3. New continuous consecution conditions

In this section we treat discrete transitions by extending and adapting our previous work on loop invariant generation for discrete programs [RMM08a, RMM08b, RMM10]. We also consider discrete transitions that are part of connected components and circuits, thus generalizing the case of simple propagations.

First, we show how to encode continuous consecution conditions.

Definition 6 Consider a transition system $W = \langle V, L, \mathcal{T}, l_0, \Theta \rangle$. Let $\tau = \langle l_i, l_j, \rho_\tau \rangle$ be a transition in $\mathcal{T}$ and let η be an algebraic inductive map with $\eta(l_i) \equiv (P_\eta(X_1, .., X_n) = 0)$ and $\eta(l_j) \equiv (P'_\eta(X_1, .., X_n) = 0)$ where P_η is a multivariate polynomial in $\mathbb{R}[X_1, .., X_n]$ such that it has null values at l_i and at l_j, i.e., before and after taking the transition. Note that this does not imply that P_η is the null polynomial. We identify the following notions when encoding *continuous consecution conditions*:

- We say that η satisfies a *Fractional-scale consecution for* τ if and only if there exists a multivariate fractional $\frac{T}{Q}$ such that $\rho_\tau \models (P_\eta(X'_1, .., X'_n) - \frac{T}{Q}P_\eta(X_1, .., X_n) = 0)$. We also say that P_η is a $\frac{T}{Q}$-scale discrete invariant.
- We say that η satisfies a *Polynomial-scale consecution for* τ if and only if there exists a multivariate polynomial T such that $\rho_\tau \models (P_\eta(X'_1, .., X'_n) - TP_\eta(X_1, .., X_n) = 0)$. We also say that P_η is a polynomial-scale and a T-scale discrete invariant.
- We say that η satisfies a *Constant-scale consecution for* τ if and only if there exists a constant λ such that $\rho_\tau \models (P_\eta(X'_1, .., X'_n) - \lambda\, P_\eta(X_1, .., X_n) = 0)$. We also say that P_η is a constant-scale, or a λ-scale discrete invariant. □

Constant-scale consecution encodes the fact that the numerical value of the polynomial P_η, associated with assertion $\eta(l_i)$, is given by λ times its numerical value throughout the transition τ. *Polynomial-scale* consecution encodes the fact that the numerical value of the polynomial P_η, associated with assertion $\eta(l_i)$, is given by T times its numerical value throughout the transition τ, where T is a polynomial in $\mathbb{R}[X_1, \ldots, X_n]$. Such T polynomials can be understood as *template multiplicative factors*, that is, they are polynomials with unknown coefficients. We are able to handle the general case when the loop describes a multivariate fractional system with *Fractional-scale* consecution. *Fractional-scale* consecution encodes the fact that the numerical value of the polynomial P_η, associated with assertion $\eta(l_i)$, is given by $\frac{T}{Q}$ times its numerical value throughout the transition τ. The fractional $\frac{T}{Q}$ can contain unknown coefficients. As can be seen, the consecution conditions are relaxed when going from constant to fractional scaling.

4. Discrete transition and affine systems

In this section we treat constant-scale consecution encodings. Consider a transition systems corresponding to the loop $\tau = \langle l_i, l_i, \rho_\tau\rangle$ and its affine transition relation

$$\rho_\tau \equiv \begin{bmatrix} X'_1 = L_1(X_1, \ldots, X_n) \\ \vdots \\ X'_n = L_n(X_1, \ldots, x_n) \end{bmatrix}. \qquad (1)$$

Here, the loop instructions are affine or linear forms $L_i(X_1, \ldots, X_n) = \sum_{k=1}^{n} c_{i,k-1}X_k + c_{i,k}$, $1 \le i \le n$.

4.1. Generating λ-scale invariants

We have the following λ-scale invariant characterization.

Theorem 1 *Consider a transition system corresponding to a loop* τ *as described in Eq. (1). A polynomial* Q *in* $\mathbb{R}[X_1, .., X_n]$ *is a* λ*-scale invariant for constant-scale consecution with* parametric constant $\lambda \in \mathbb{R}$ *for* τ *if and only if* $Q(L_1(X_1, .., X_n), .., L_n(X_1, .., X_n)) = \lambda\, Q(X_1, .., X_n)$. □

Let the degree of $Q \in \mathbb{R}[X_1, .., X_n]$ be r. We show that for good choices of λ there always exists such a λ-invariant that is also *non-trivial*. We note that $Q(L_1(X_1, .., X_n), .., L_n(X_1, .., X_n))$ is also of degree r because all L_i's are of degree 1. Recasting the situation and ρ_τ into linear algebra, consider the morphism

$$\mathscr{M} : \begin{cases} \mathbb{R}_r[X_1, \ldots, X_n] \to \mathbb{R}_r[X_1, \ldots, X_n] \\ Q(X_1, \ldots, X_n) \mapsto Q(L_1(X_1, .., X_n), \ldots, L_n(X_1, .., X_n)). \end{cases}$$

This is indeed an endomorphism because all L_i's are of degree 1. Let M be its matrix representation in the canonical basis of $\mathbb{R}_r[X_1, ., X_n]$. First, we show how we can build matrix M.

Example 2 Consider the following loop $\rho_\tau = \begin{bmatrix} x_1' = 2x_1 + x_2 + 1 \\ x_2' = 3x_2 + 4 \end{bmatrix}$. We have two polynomials of degree 1, in two variables. They are $L_1(x_1, x_2) = 2x_1 + x_2 + 1$, and $L_2(x_1, x_2) = 3x_2 + 4$. Consider the associated endomorphism $\mathscr{M}$ from $\mathbb{R}_2[x_1, x_2]$ to $\mathbb{R}_2[x_1, x_2]$. We want to obtain an associated matrix M for it. For that, we can use $B_1 = (x_1^2, x_1x_2, x_2^2, x_1, x_2, 1)$ as a basis for $\mathbb{R}_2[x_1, x_2]$ and compute $\mathscr{M}(P)$ for all elements P in B_1, expressing the results in the same basis. For the first column of M we consider $P(x_1, x_2) = x_1^2$ as the first element of B_1, and compute $\mathscr{M}(P) = P(L_1(x_1, x_2), L_2(x_1, x_2))$, which is expressed in B_1 as

$$\mathsf{M}\ (x_1^2) = 4\,x_1^2 + 4\,x_1x_2 + 1\,x_2^2 + 4\,x_1 + 2\,x_2 + 1 \times 1$$

$$M = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 0 & 0 & 0 & 0 \\ 1 & 3 & 9 & 0 & 0 & 0 \\ 4 & 8 & 0 & 2 & 0 & 0 \\ 2 & 7 & 24 & 1 & 3 & 0 \\ 1 & 4 & 16 & 1 & 4 & 1 \end{pmatrix}.$$

This concludes the example. □

Now, let $Q \in \mathbb{R}[X_1, .., X_n]$ be a λ-scale invariant for constant-scale consecution with *parametric constant* $\lambda \in \mathbb{R}$ for a given system defined by $L_1, .., L_n \in \mathbb{R}[X_1, .., X_n]$. By Theorem 1, we have

$$Q(L_1(X_1, \ldots, X_n), .., L_n(X_1, \ldots, X_n)) = \lambda Q(X_1, \ldots, X_n).$$

Using the associated endomorphism $\mathscr{M}$, we have:

$$\begin{aligned} Q(L_1(X_1, \ldots, X_n), .., L_n(X_1, \ldots, X_n)) = \lambda Q(X_1, \ldots, X_n) \quad &\Leftrightarrow \\ \mathscr{M}(Q) = \lambda Q \quad &\Leftrightarrow \\ \mathscr{M}(Q) = \lambda \mathscr{I}(Q) \quad &\Leftrightarrow \\ (\mathscr{M} - \lambda \mathscr{I})(Q) = 0_{\mathbb{R}[X_1,..,X_n]} \quad &\Leftrightarrow \\ Q \in Ker(M - \lambda I), \quad & \end{aligned}$$

where $\mathscr{I}$ is the identity endomorphism and I is the associated identity matrix in $\mathbb{R}_r[X_1, ..X_n]$. Hence, λ must be an eigenvalue of M if we want to find a non null λ-invariant whose coefficients will be those of an eigenvector.

We can now state the following theorem.

Theorem 2 *A polynomial Q of $\mathbb{R}_r[X_1, .., X_n]$ is λ-invariant for constant-scale consecution if and only if there exists an eigenvalue λ of M such that Q belongs to the eigenspace corresponding to λ.* □

We also notice that, by construction, the last column of M is always $(0, \ldots, 0, 1)^\top$. Thus 1 is always an eigenvalue of M with a corresponding eigenvector which leads to the *trivial* λ-invariant $Q(X_1, .., X_n) = a$, where a is the coefficient of the constant term. Eigenvalue 1 always gives the constant polynomial as a λ-invariant, but it might give better invariants for other eigenvectors if $dim(Ker(M - \lambda I)) \geq 2$, as we will see in the sequel.

Example 3 Looking at the eigenvalues of the matrix M in Example 2, if we fix λ to be 4 we get that the corresponding eigenspace is generated by the vector $(1, -2, 1, -6, 6, 9)^\top$. Interpreted in the canonical basis of $\mathbb{R}[x_1, x_2]$, the associated 4-invariant is $Q(x_1, x_2) = 1{x_1}^2 - 2x_1x_2 + {x_2}^2 - 6x_1 + 6x_2 + 9$. □

We first treat the general case where the transition system has only two variables. We will look for a λ-invariant Q of degree two. Let

$$\rho_\tau = \begin{bmatrix} x_1' = c_{1,0}x_1 + c_{1,1}x_2 + c_{1,2} \\ x_2' = c_{2,0}x_1 + c_{2,1}x_2 + c_{2,2} \end{bmatrix}.$$

Recall that we must solve the equation $Q(c_{1,0}X_1 + c_{1,1}X_2 + c_{1,2}, c_{2,0}X_1 + c_{2,1}X_2 + c_{2,2}) = \lambda Q(X_1, X_2)$. Thus, for M we get the following matrix:

$$\begin{pmatrix} {c_{1,0}}^2 & c_{1,0}c_{2,0} & {c_{2,0}}^2 & 0 & 0 & 0 \\ 2c_{1,0}c_{1,1} & c_{1,0}c_{2,1} + c_{1,1}c_{2,0} & 2c_{2,0}c_{2,1} & 0 & 0 & 0 \\ {c_{1,1}}^2 & c_{1,1}c_{2,1} & {c_{2,1}}^2 & 0 & 0 & 0 \\ 2c_{1,0}c_{1,2} & c_{1,0}c_{2,2} + c_{1,2}c_{2,0} & 2c_{2,0}c_{2,2} & c_{1,0} & c_{2,0} & 0 \\ 2c_{1,1}c_{1,2} & c_{1,1}c_{2,2} + c_{1,2}c_{2,1} & 2c_{2,1}c_{2,2} & c_{1,1} & c_{2,1} & 0 \\ {c_{1,2}}^2 & c_{1,2}c_{2,2} & {c_{2,2}}^2 & c_{1,2} & c_{2,2} & 1 \end{pmatrix}.$$

We see that the last column is as predicted, plus the matrix is block diagonal. Thus its characteristic polynomial is $P(\lambda) = (1 - \lambda)P_1(\lambda)P_2(\lambda)$, with P_1 being the characteristic polynomial of

$$\begin{pmatrix} c_{1,0} & c_{2,0} \\ c_{1,1} & c_{2,1} \end{pmatrix},$$

and P_2 being the characteristic polynomial of

$$\begin{pmatrix} {c_{1,0}}^2 & c_{1,0}c_{2,0} & {c_{2,0}}^2 \\ 2c_{1,0}c_{1,1} & c_{1,0}c_{2,1} + c_{1,1}c_{2,0} & 2c_{2,0}c_{2,1} \\ {c_{1,1}}^2 & c_{1,1}c_{2,1} & {c_{2,1}}^2 \end{pmatrix}.$$

Here P_2 is of degree 3 and has at least one real root. This root can be computed by the Lagrange resolvent method. Choosing λ to be this root, the corresponding eigenvectors will give non-trivial λ-invariants of degree two, since at least one of the coefficients of the monomials x_1^2, x_1x_2 and x_2^2 must be non null for such an eigenvector.

Corollary 1 *Let M be the matrix introduced in this section. The problem of finding a* non-trivial λ-invariant *is* decidable *if one of the following assertions is true:*

- *M is block triangular (with 4×4 blocks or less),*
- *The eigenspace associated with eigenvalue 1 is of dimension greater than 1.* □

4.2. Intersection with initial hyperplanes

Let $Q \in \mathbb{R}_r[X_1, .., X_n]$ be a λ-invariant for constant-scale consecution, that is,

$$Q(L_1(X_1, .., X_n), .., L_n(X_1, .., X_n)) = \lambda Q(X_1, .., X_n).$$

Now let $u_1, \ldots, u_n$ be the initial values of $X_1, \ldots, X_n$. For the initial step we need $Q(u_1, \ldots, u_n) = 0$. We have $P \mapsto P(u_1, \ldots, u_n)$ as a linear form in $\mathbb{R}_r[X_1, \ldots, X_n]$. Hence initial values correspond to a hyperplane in $\mathbb{R}_r[X_1, \ldots, X_n]$, given by the kernel of $P \mapsto P(u_1, \ldots, u_n)$. If we add the initiation step, $Q(X_1, \ldots, X_n) = 0$ will be an inductive invariant (see Definition 4) if and only if there exists an eigenvalue λ of M such that Q belongs to the intersection of the eigenspace corresponding to λ and the hyperplane $Q(u_1, \ldots, u_n) = 0$.

Theorem 3 *A polynomial Q in $\mathbb{R}_r[X_1, .., X_n]$ is an inductive invariant for the affine loop (see Definition 5) with initial values $(u_1, \ldots, u_n)$ if and only if there is an eigenvalue λ of M such that Q is in the intersection of the eigenspace of λ and the hyperplane $Q(u_1, \ldots, u_n) = 0$.* □

In the following corollary, we state an important result.

Corollary 2 *There will be a non-null polynomial invariant for any given initial values if and only if there exists an eigenspace of M with dimension at least 2.* □

Example 4 We return to running Example 2. Matrix M has 6 distinct eigenvalues, and so the corresponding eigenspaces are of dimension 1. We denote by E_λ the eigenspace corresponding to λ. Then E_4 has a basis $(1, -2, 1, -6, 6, 9)^\top$, E_6 has a basis $(0, 1, -1, 2, -5, 6)^\top$, E_9 has a basis $(0, 0, 1, 0, 4, 4)^\top$, E_2 has a basis $(0, 0, 0, 1, -1, -3)^\top$, E_3 has a basis $(0, 0, 0, 0, 1, 2)^\top$, and E_1 has a basis $(0, 0, 0, 0, 0, 1)^\top$. Also, suppose that the initiation step is given by $(x_1 = 0, x_2 = -2)$, i.e., $(u_1, u_2) = (0, 2)$, which corresponds to the hyperplane $Q(0, 2) = 0$ in $\mathbb{R}_2[x_1, x_2]$.

We start with simple initial conditions and consider general conditions in the sequel. Theorem 3 applies, and since it is clear that $(0, 0, 1, 0, 4, 4)^\top$ belongs to the hyperplane, we get $x_2{}^2 + 4x_2 + 4 = 0$ is an inductive invariant for that loop with these specific initial conditions. □

Example 5 We study the following transition system [SSM04b], corresponding to the multiplication of 2 numbers, and where the transition is $\tau = \langle l_i, l_i, \rho_\tau \rangle$, with

$$\rho_\tau = \begin{bmatrix} s' = s + i \\ j' = j - 1 \\ i' = i \\ j_0' = j_0 \end{bmatrix}.$$

We need to find a λ such that $Q(s + i, j + 1, i, j_0) = \lambda Q(s, j, i, j_0)$.

- *Step 1*: We build the associated matrix M:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1
\end{pmatrix}.$$

- *Step 2*: We compute the eigenvectors which will provide us with a basis for non-trivial λ-invariants. Here, an evident eigenvalue is 1.
- *Step 3*: It is clear, in view of the matrix M, that $dim(Ker(M - I)) \geq 2$. As the eigenspace associated to eigenvalue 1 is of dimension 2, Corollary 2 applies. For example, the vector

  $(0, 0, 0, 0, 0, 1, 0, 0, -1, 0, 1, 0, 0, 0, 0)^\top$

  is the eigenvector corresponding to the λ-invariant $s + ji - ij_0$.

Note that without computing Gröbner bases or performing quantifier elimination, we found the invariant $s + ji - ij_0 = 0$ obtained by Sankaranarayanan et al. in [SSM04b]. The consecution scale technique will give a non-null invariant whatever the initial values are, and this explains why a non-trivial invariant was found in that work. □

4.3. Limits of constant-scale consecution

Here we consider an algebraic transition relation where the instructions are described by polynomials with degree greater than 1.

Example 6 Consider the following loop: $\rho_\tau \equiv \begin{bmatrix} x' = x(y + 1) \\ y' = y^2 \end{bmatrix}$. At step k of the iteration, this loop computes the sum $1 + y + \cdots + y^{2^k - 1}$. Let $P(x, y) = a_0x^2 + a_1xy + a_2y^2 + a_3x + a_4y + a_5$ be a candidate λ-invariant. With the Gröbner Bases $\{x' - x(y + 1), y' - y^2\}$, and with the total-degree lexicographic ordering given by the precedence $x' > y' > x > y$, we can get the loop ideal of $\mathbb{K}[x', y', x, y]$. Modulo this loop ideal, we have $P(x', y') = P(x(y + 1), y^2)$. Put $P'(x, y) = P(x(y + 1), y^2)$. After expanding we get $P'(x, y) = a_0x^2y^2 + a_1xy^3 + a_2y^4 + 2a_0x^2y + a_1xy^2 + a_0x^2 + a_3xy + a_4y^2 + a_3x + a_5$. If we try a constant-scale consecution with parameter λ we obtain:

$$\begin{array}{lll} a_0 = 0 & a_1 = 0 & a_3 = \lambda\, a_3 \\ a_1 = 0 & a_0 = \lambda\, a_0 & \lambda\, a_4 = 0 \\ a_2 = 0 & a_3 = \lambda\, a_1 & a_5 = \lambda\, a_5 \\ 2a_0 = 0 & a_4 = \lambda\, a_2. & \end{array}$$

After simplifications, we get $a_0 = a_1 = a_2 = a_3 = a_4 = 0$ and $a_5 = \lambda\, a_5$. If $\lambda \neq 1$ then $a_5 = 0$, which leads to a *null* invariant. Otherwise, $\lambda = 1$ and we obtain the *constant* invariant a_5. Also, the initial condition implies that the constant invariant a_5 is null. So, using a constraint-based approach with constant-scaling [SSM04b] we can obtain only constant or null, i.e. *trivial*, invariants. □

In the following section, we show how we handle this problem.

5. Algebraic discrete transition systems

In this section, we approach non-linear discrete systems.

5.1. T-scale invariant generation

Consider an algebraic transition system: $\rho_\tau \equiv \begin{bmatrix} X_1' = P_1(X_1, .., X_n) \\ \vdots \\ X_n' = P_n(X_1, .., X_n) \end{bmatrix}$, where the loop updates can be represented using polynomials in $\mathbb{R}[X_1, .., X_n]$ of the forms $P_i(X_1, .., X_n) = \sum_{i_1,..,i_n} a_{i_1,...,i_n} {X_1}^{i_1} \ldots {X_n}^{i_n}$, where the coefficients $a_{i_1,\ldots,i_n}$ are in $\mathbb{R}$. We have the following T-scale discrete invariant characterization.

Theorem 4 *A polynomial Q in $\mathbb{R}[X_1, .., X_n]$ is a T-scale discrete invariant for polynomial-scale consecution with a* parametric polynomial $T \in \mathbb{R}[X_1, \ldots, X_n]$ *for τ if and only if*

$$Q(P_1(X_1, .., X_n), .., P_n(X_1, .., X_n)) = T(X_1, .., X_n)Q(X_1, .., X_n).$$

□

Example 7 Reconsider Example 6. We now take $(y = y_0, x = 1)$ as initial values. We want to obtain a *polynomial scale consecution* with a parametric polynomial $T(x, y) = b_0y^2 + b_1x + b_2y + b_3$. We thus obtain $P'(s, x) = (b_0y^2 + b_1x + b_2y + b_3)P(x, y)$. In other words, we obtain the following multi-parametric linear system with parameters b_0, b_1, b_2, b_3:

$$\begin{array}{lll} a_0 = b_0 a_0 & 0 = b_2 a_5 + b_3 a_4 & a_3 = b_1 a_4 + b_2 a_3 + b_3 a_1 \\ a_1 = b_0 a_1 & 0 = b_0 a_4 + b_2 a_2 & a_4 = b_0 a_5 + b_2 a_4 + b_3 a_2 \\ a_2 = b_0 a_2 & a_3 = b_1 a_5 + b_3 a_3 & a_1 = a_3 b_0 + b_1 a_2 + b_2 a_1 \\ a_5 = b_3 a_5 & a_0 = b_1 a_3 + b_3 a_0 & \\ 0 = b_1 a_0 & 2a_0 = b_1 a_1 + b_2 a_0. & \end{array}$$

We now describe a decision procedure for choosing parameter values. Consider the first three equations and choose $b_0 = 1$. In this way we aim at a high degree invariant for, otherwise, the coefficients a_0, a_1, a_2 of the highest degree terms would be null. Then, we are lead to another system with $b_1 a_0 = 0$. For the same reason, choose $b_1 = 0$. Then we have $b_2 a_0 = 2a_0$. As a direct consequence, b_2 is set to 2. Since equation $b_3 a_0 = a_0$ is in the resulting system, b_3 is set to 1. Finally, we obtain the following system:

$$\begin{aligned} a_3 + a_1 &= 0 \\ a_4 + 2a_2 &= 0 \\ a_2 - a_5 &= 0. \end{aligned}$$

Having less equations than variables, we will have a *non-trivial* solution for generating of T-invariants. Now, we consider the hyperplane corresponding to the initial values, that is, $a_2{y_0}^2 + (a_1 + a_4)y_0 + a_0 + a_1 + a_5 = 0$. As there are six variables and four equations, we will have again a *non-trivial* solution. A possible solution is the vector $(y_0(1-y_0), 1, 1, -1, -2, 1)^\top$. So, $y_0(1-y_0)x^2 + xy + y^2 - x - 2y + 1 = 0$ is an invariant. Note that $T(x, y) = y^2 + y + 1$. □

Remark 1 That is a simple constraint-based procedure, which can fail in more complex cases. Shortly, we will present a superior technique, from a more encompassing point of view. □

5.2. A general theory for discrete transitions and polynomial systems

If $Q \in \mathbb{R}[X_1, .., X_n]$ is of degree r and the maximal degree of the P_i's is d, then we are looking for a T of degree $e = dr - r$. Write its ordered coefficients as $\lambda_0, \ldots, \lambda_s$, with $s + 1$ being the number of monomials of degree inferior to e.

Let M be the matrix, in the canonical basis of $\mathbb{R}_r[X_1, .., X_2]$ and $\mathbb{R}_{dr}[X_1, .., X_n]$, of the morphism

$$\mathscr{M} : \begin{cases} \mathbb{R}_r[X_1, \ldots, X_n] \to \mathbb{R}_{dr}[X_1, \ldots, X_n] \\ Q(X_1, \ldots, X_n) \mapsto Q(P_1(X_1, .., X_n), \ldots, P_n(X_1, .., X_n)). \end{cases}$$

Let L be the matrix, in the canonical basis of $\mathbb{R}_r$ and $\mathbb{R}_{dr}$, of the morphism

$$\mathscr{L} : \begin{cases} \mathbb{R}_r[X_1, \ldots, X_n] \to \mathbb{R}_{dr}[X_1, \ldots, X_n] \\ \qquad P \mapsto TP. \end{cases}$$

Matrix L has a very simple form: its non zero coefficients are the λ_i's, and it has a natural block decomposition. Now let $Q \in \mathbb{R}[X_1, .., X_n]$ be a T-scale discrete invariant for a transition relation defined by the P_i's. Then,

$$\begin{aligned} Q(P_1(X_1, .., X_n), .., P_n(X_1, .., X_n)) &= T(X_1, .., X_n)Q(X_1, .., X_n) && \Leftrightarrow \\ \mathscr{M}(Q) &= \mathscr{L}(Q) && \Leftrightarrow \\ (\mathscr{M} - \mathscr{L})(Q) &= 0_{\mathbb{R}[X_1,..,X_n]} && \Leftrightarrow \\ Q &\in Ker(M - L). \end{aligned}$$

A T-scale discrete invariant is nothing else than a vector in the kernel of $M - L$. Our problem is equivalent to finding a L such that $M - L$ has a non-trivial kernel.

Theorem 5 *Consider M as described above. Then, there will be a T-scale discrete invariant if and only if there exists a matrix L, corresponding to $P \mapsto TP$, such that $M - L$ has a nontrivial kernel. Further, any vector in the kernel of $M - L$ will give rise to a T-scale invariant.* □

We denote by $v(r)$ the dimension of $\mathbb{R}_r[X_1, \ldots, X_n]$. Again, the last column of M is $(0, \ldots, 0, 1)^\top$. The last column of L is $(0, .., 0, \lambda_0, .., \lambda_s)^\top$. Hence, choosing every λ_i to be zero, except for $\lambda_s = 1$, the last column of $M - L$ will be null. With this choice of L (or $T = 1$), we get at least T-invariants corresponding to constant polynomials. Now, $M - L$ having a non-trivial kernel is equivalent to its rank being less than the dimension $v(r)$ of $\mathbb{R}_r[X_1, \ldots, X_n]$. This is equivalent to the fact that each $v(r) \times v(r)$ sub-determinant of $M - L$ is equal to zero [Lan02]. Those determinants are polynomials in variables $(\lambda_0, \lambda_1, \ldots, \lambda_s)$, which we will denote by $V_1(\lambda_0, \lambda_1, \ldots, \lambda_s), \ldots, V_s(\lambda_0, \lambda_1, \ldots, \lambda_s)$.

Theorem 6 *There is a non-trivial T-scale invariant if and only if the polynomials* $(V_1, .., V_s)$ *admit a common root, other than the trivial one* $(0, \ldots, 0, 1)$. □

Remark 2 This theorem provides us with important existence results. But there is a more practical way of computing invariant ideals without computing common roots and sub-determinants. We will examine that in the next section. □

We first study the general case of degree two algebraic transition systems with two variables in the loop. Such transition systems have the form: $\rho_\tau \equiv \begin{bmatrix} x' = c_0x^2 + c_1xy + c_2y^2 + c_3x + c_4y + c_5 \\ y' = d_0x^2 + d_1xy + d_2y^2 + d_3x + d_4y + d_5 \end{bmatrix}$. In this case, matrices M and L will be as follows:

$$M = \begin{pmatrix} {c_0}^2 & c_0d_0 & {d_0}^2 & 0 & 0 & 0 \\ 2c_0c_1 & c_0d_1 + c_1d_0 & 2d_0d_1 & 0 & 0 & 0 \\ 2c_0c_2 + {c_1}^2 & c_0d_2 + c_1d_1 + c_2d_0 & 2d_0d_2 + {d_1}^2 & 0 & 0 & 0 \\ 2c_1d_1 & c_1d_2 + c_2d_1 & 2d_1d_2 & 0 & 0 & 0 \\ {c_2}^2 & c_2d_2 & {d_2}^2 & 0 & 0 & 0 \\ 2c_0c_3 & c_0d_3 + c_3d_0 & 2d_0d_3 & 0 & 0 & 0 \\ 2(c_0c_4 + c_1c_3) & c_0d_4 + c_1d_3 + c_3d_1 + c_4d_0 & 2(d_0d_4 + d_1d_3) & 0 & 0 & 0 \\ 2(c_1c_4 + c_2c_3) & c_1d_4 + c_2d_3 + c_3d_2 + c_4d_1 & 2(d_1d_4 + d_2d_3) & 0 & 0 & 0 \\ 2c_2c_4 & c_2d_4 + c_4d_2 & 2d_2d_4 & 0 & 0 & 0 \\ 2c_0c_5 + {c_3}^2 & c_0d_5 + c_3d_3 + c_5d_0 & 2d_0d_5 + {d_3}^2 & c_0 & d_0 & 0 \\ 2(c_1c_5 + c_3c_4) & c_1d_5 + c_3d_4 + c_4d_3 + c_5d_1 & 2(d_1d_5 + d_3d_4) & c_1 & d_1 & 0 \\ 2c_2c_5 + {c_4}^2 & c_2d_5 + c_4d_4 + c_5d_2 & 2d_2d_5 + {d_4}^2 & c_2 & d_2 & 0 \\ 2c_3c_5 & c_3d_5 + c_5d_3 & 2d_3d_5 & c_3 & d_3 & 0 \\ 2c_4c_5 & c_4d_5 + c_5d_4 & 2d_4d_5 & c_4 & d_4 & 0 \\ {c_5}^2 & c_5d_5 & {d_5}^2 & c_5 & d_5 & 1 \end{pmatrix} L = \begin{pmatrix} \lambda_0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_1 & \lambda_0 & 0 & 0 & 0 & 0 \\ \lambda_2 & \lambda_1 & \lambda_0 & 0 & 0 & 0 \\ 0 & \lambda_2 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 & 0 & \\ \lambda_3 & 0 & 0 & \lambda_0 & 0 & 0 \\ \lambda_4 & \lambda_3 & 0 & \lambda_1 & \lambda_0 & 0 \\ 0 & \lambda_4 & \lambda_3 & \lambda_2 & \lambda_1 & 0 \\ 0 & 0 & \lambda_4 & 0 & \lambda_2 & 0 \\ \lambda_5 & 0 & 0 & \lambda_3 & 0 & \lambda_0 \\ 0 & \lambda_5 & 0 & \lambda_4 & \lambda_3 & \lambda_1 \\ 0 & 0 & \lambda_5 & 0 & \lambda_4 & \lambda_2 \\ 0 & 0 & 0 & \lambda_5 & 0 & \lambda_3 \\ 0 & 0 & 0 & 0 & \lambda_5 & \lambda_4 \\ 0 & 0 & 0 & 0 & 0 & \lambda_5 \end{pmatrix}.$$

For the rank of $M - L$ to be less than 6, one has to calculate each 6×6 sub-determinant obtained by canceling 9 lines of $M - L$. They will be polynomials of degree less than 6 in the variables $(\lambda_0, \ldots, \lambda_5)$. In this way $M - L$ will be of degree less than 6 if and only if $(\lambda_0, \ldots, \lambda_5)$ are roots of each of those polynomials. In many cases, it is easy to find a matrix L such that $M - L$ has a non-trivial kernel. We describe and deal with several decidable classes (see Table 1b line 3 and the lines 13–20). The following important remark make clear the advances reached when comparing this approach to related constraint-based methods.

Remark 3 The unknown coefficients $(\lambda_0, \ldots, \lambda_s)$ appearing in the matrix L, used to approximate the consecution conditions, are assigned in order to insure that the nullspaces generated are not trivial ones. In fact, these unknown coefficients do not have the same roles as the templates used in constraint-based approaches. In our method, these parameters do not take part in a constraint solving problem. Instead, they allow us to obtain a sufficiently precise approximation to the consecution condition in order to guarantee the existence and the computation of vector spaces of T-invariants, that is, nullspaces. □

5.3. Generating invariant ideals with an initiation step

Consider initial values given by unknown parameters $(X_1 = u_1, \ldots, X_n = u_n)$. The initial step defines, on $\mathbb{R}_r[x_1, \ldots, x_n]$, a linear form $P \mapsto P(u_1, \ldots, u_n)$. Hence, initial values correspond to a hyperplane in $\mathbb{R}_r[X_1, .., X_n]$, given by the kernel of $P \mapsto P(u_1, \ldots, u_n)$, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] \mid Q(u_1, \ldots, u_n) = 0\}$.

Theorem 7 *Let Q be in $\mathbb{R}_r[X_1, .., X_n]$. Then Q is an inductive invariant for the transition system with initial values $(u_1, .., u_n)$ if and only if there exists a matrix $L \neq 0$, i.e, one of $P \mapsto TP$, corresponding to T in $\mathbb{R}_e[X_1, .., X_n]$, such that Q is in the intersection of $Ker(M - L)$ and the hyperplane given by the initial values $Q(u_1, \ldots, u_n) = 0$. The invariants will correspond to vectors in the intersection.* □

Now, if $Dim(Ker(M - L)) \geq 2$ then $Ker(M - L)$ will intersect any initial (semi-)hyperplane. We can state the following corollary, important in practice.

Corollary 3 *There are non-trivial invariants for any given initial values if and only if there exists a matrix L such that $Ker(M - L)$ has dimension at least 2. The basis of $Ker(M - L)$ being a basis for non-trivial invariants.* □

There are non-trivial invariants for any given initial values if and only if there exists a matrix L, corresponding to multiplicative template in T, such that $Ker(M - L)$ has dimension at least 2.

5.4. Example

Consider the following transition: $\tau = \left\langle l_i, l_j, \rho_\tau \equiv \begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix} \right\rangle$.

- *Step 1*: We build the matrix $M - L$. The maximal degree of ρ_τ is $d = 2$, and so the T-scale invariant will be of degree $r = 2$. Also, T is of degree $e = dr - r = 2$ and we write $\lambda_0, \dots, \lambda_5$ as its ordered coefficients. Then its canonical form is $T = \lambda_0 x^2 + \lambda_1 xy + \lambda_2 y^2 + \lambda_3 x + \lambda_4 y + \lambda_5$. Consider the associated morphisms $\mathscr{M}$ and $\mathscr{L}$ from $\mathbb{R}_2[x, y]$ to $\mathbb{R}_4[x, y]$. Using the basis $C_1 = (x^2, xy, y^2, x, y, 1)$ of $\mathbb{R}_2[x, y]$ and the basis $C_2 = (x^4, yx^3, y^2x^2, y^3x, y^4, x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1)$ of $\mathbb{R}_4[x, y]$, our algorithm compute the matrix $M - L$ as

$$M - L_{(\lambda_0,\dots,\lambda_5)} = \begin{pmatrix} -\lambda_0 & 0 & 0 & 0 & 0 & 0 \\ -\lambda_1 & -\lambda_0 & 0 & 0 & 0 & 0 \\ 1-\lambda_2 & -\lambda_1 & -\lambda_0 & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & -\lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda_2 & 0 & 0 & 0 \\ -\lambda_3 & 0 & 0 & -\lambda_0 & 0 & 0 \\ 2-\lambda_4 & -\lambda_3 & 0 & -\lambda_1 & -\lambda_0 & 0 \\ 0 & 1-\lambda_4 & -\lambda_3 & -\lambda_2 & -\lambda_1 & 0 \\ 0 & 0 & -\lambda_4 & 0 & -\lambda_2 & 0 \\ 1-\lambda_5 & 0 & 0 & -\lambda_3 & 0 & -\lambda_0 \\ 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_3 & -\lambda_1 \\ 0 & 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_2 \\ 0 & 0 & 0 & 1-\lambda_5 & 0 & -\lambda_3 \\ 0 & 0 & 0 & 0 & -\lambda_5 & -\lambda_4 \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_5 \end{pmatrix}.$$

- *Step 2*: We now reduce the rank of $M - L$ by assigning values to the λ_i's. Our procedure fixes $\lambda_0 = \lambda_1 = \lambda_3 = 0$, $\lambda_2 = \lambda_5 = 1$ and $\lambda_4 = 2$, so that $T(x, y) = y^2 + 2y + 1$. The first column of $M - L$ becomes zero and the second column is equal to the fourth. Hence, the rank of $M - L$ is less than 4 and its kernel has dimension at least 2. Any vector in this kernel will be a T-invariant.
  Before, proceeding to Step 3 we give more details on our *rank reduction procedure* which allows us to choose the coefficients $\lambda_0, \dots, \lambda_4$ and λ_5 such that the matrix $M - L$ does not have a trivial kernel. Taking into consideration the specific type of matrix we are manipulating, we are going to determine for which values of $\lambda_0, \dots, \lambda_4$ and λ_5, the rank of $M - L$ is minimal. We proceed by an analysis over the top non-zero elements of the columns of $M - L$ and the possible values that could be chosen for the parameters in order to decrease the actual rank. At each step of the assignments, we will echelon the matrix by making the highest term of one column to vanish. For that, we index the matrix $M - L$ as $M - L_{(\lambda_0,\dots,\lambda_5)}$, in order to keep track of the assignment and rank obtained during the procedure.
  Looking at the top non-zero elements of $M - L_{(\lambda_0,\dots,\lambda_5)}$, consider first the parameter λ_0. If λ_0 is not zero, we obtain an echelon form matrix of rank 6, which is maximal. Thus λ_0 is fixed to 0. Now, considering the top non-zero elements of the columns of $M - L_{(0,\lambda_1,\dots,\lambda_5)}$, the procedure fixes λ_1 to 0 for the same reason. We obtain the following matrix:

$$M - L_{(0,0,\lambda_2,\dots,\lambda_5)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1-\lambda_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda_2 & 0 & 0 & 0 \\ -\lambda_3 & 0 & 0 & 0 & 0 & 0 \\ 2-\lambda_4 & -\lambda_3 & 0 & 0 & 0 & 0 \\ 0 & 1-\lambda_4 & -\lambda_3 & -\lambda_2 & 0 & 0 \\ 0 & 0 & -\lambda_4 & 0 & -\lambda_2 & 0 \\ 1-\lambda_5 & 0 & 0 & -\lambda_3 & 0 & 0 \\ 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_3 & 0 \\ 0 & 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_2 \\ 0 & 0 & 0 & 1-\lambda_5 & 0 & -\lambda_3 \\ 0 & 0 & 0 & 0 & -\lambda_5 & -\lambda_4 \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_5 \end{pmatrix}.$$

Again, we look at the non-zero top column elements with λ_2. If λ_2 is not in $\{1, 0\}$, we obtain an echelon form matrix of rank 6, which is maximal. Thus we need to consider the case where λ_2 is assigned a value in $\{1, 0\}$. Hence λ_2 is first assigned to 1. Then λ_3 has to be 0, otherwise the rank will be maximal. The procedure is now working with:

$$M - L_{(0,0,1,0,\lambda_4,\lambda_5)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2-\lambda_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\lambda_4 & 0 & -1 & 0 & 0 \\ 0 & 0 & -\lambda_4 & 0 & -1 & 0 \\ 1-\lambda_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_5 & 0 & 1-\lambda_4 & 0 & 0 \\ 0 & 0 & -\lambda_5 & 0 & 1-\lambda_4 & -1 \\ 0 & 0 & 0 & 1-\lambda_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_5 & -\lambda_4 \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_5 \end{pmatrix}.$$

In order to reduce its rank by making the highest term of a column vanish, λ_4 is assigned a value in $\{2, 1, 0\}$. Then, λ_4 is first fixed to 2 which implies that λ_5 needs to be assigned to 1. We conclude that with $(\lambda_0, \ldots, \lambda_4, \lambda_5) = (0, 0, 1, 0, 2, 1)$ the matrix $M - L$ does not have a trivial kernel and $T(x, y) = y^2 + 2y + 1$ can be used to generate a vector space of T-invariants. The procedure does not stop here. It continues to consider the other possibilities for λ_2, λ_4 and λ_5, thus generating more polynomials for approximating scaling consecution, leading to other vector spaces of polynomial scale and inductive invariants. With $\lambda_2 = 1$ and $\lambda_4 = 1$ or $\lambda_4 = 0$ one has to fix $\lambda_5 = 0$ and generate the polynomials $T_2(x, y) = y^2 + y$ and $T_3(x, y) = y^2$. It remains to treat the case $\lambda_2 = 0$ and the possibilities thereof. One needs to consider again $M - L_{(0,0,\lambda_2,\ldots,\lambda_5)}$. In this case, one needs again to fix λ_3 to 0. We obtain the matrix:

$$M - L_{(0,0,0,0,\lambda_4,\lambda_5)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2-\lambda_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\lambda_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\lambda_4 & 0 & 0 & 0 \\ 1-\lambda_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_5 & 0 & 1-\lambda_4 & 0 & 0 \\ 0 & 0 & -\lambda_5 & 0 & 1-\lambda_4 & 0 \\ 0 & 0 & 0 & 1-\lambda_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_5 & -\lambda_4 \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_5 \end{pmatrix}.$$

Then, looking at the top column elements of $M - L_{(0,0,0,0,\lambda_4,\ldots,\lambda_5)}$, we only have now two possibilities for λ_4: it has to be assigned a value in $\{1, 0\}$. When $\lambda_4 = 1$ we need to treat $\lambda_5 = 1$ and $\lambda_5 = 0$ leading to the polynomials $T_4(x, y) = y + 1$ and $T_5(x, y) = y$. Now, treating the other case, with $\lambda_4 = 0$, one has no choice but to assign λ_5 to 1 leading to the polynomial $T_6(x, y) = 1$.
Conclusion of Step 2: The polynomials $T, T_2, \ldots, T_6$ would guarantee a non-trivial Kernel for $M - L$. We move to Step 3 considering $T(x, y) = y^2 + 2y + 1$ because with $(\lambda_0, \ldots, \lambda_5)$, the rank of $M - L$ is minimal (i.e., null space of dimensions 3).

- *Step 3*: Now matrix $M - L$ satisfies the hypotheses of Theorem 5. So, there will always be invariants, whatever the initial values. We compute a basis of $Ker(M - L)$: `[(1,0,0,0,0,0),(0,1,0,-1,0,0),(0,0,1,0,-2,1)]`. The vectors of the basis are interpreted in the canonical basis C_1 of $\mathbb{R}_2[x, y]$, giving: $\{\{x^2, xy - x, y^2 - 2y + 1\}\}$

We have obtained an ideal for non-trivial inductive invariants. In other words, for all $G_1, G_2, G_3 \in \mathbb{R}[x, y]$, $G_1(x, y)(x^2) + G_2(x, y)(xy - x) + G_3(x, y)(y^2 - 2y + 1) = 0$ is an inductive invariant. For instance, consider the initial step $(y = y_0, x = 1)$. A possible invariant is $y_0(1 - y_0)x^2 + xy - x + y^2 - 2y + 1 = 0$. We can also consider $T_2(x, y) = y^2 + y$ and $T_4(x, y) = y + 1$ as they both provide kernels of dimension at least 2, leading to the two other vector spaces of non-trivial inductive invariants.

6. Obtaining optimal degree bounds for discrete transition systems

In order to guarantee the existence of non-trivial invariants, we look for a polynomial T such that $Ker(M-L) \neq 0$. The pseudo code depicted in Algorithm 1 illustrates the strategy. Its contribution relies on very general sufficient conditions for the existence and computation of invariants.

As input we have r, the candidate degree for the basis invariant elements, and $P_1, ..P_n$, the n polynomials given by the transition relation in the loop program. We first compute d, the maximal degree of the P_i's as can be seen by `Max_degree`$(\{P_1, \ldots, P_n\})$, at line 4. Following the instructions provided in Sect. 4 the function `Matrix_D`$(r, dr, P_1, \ldots, P_n)$ construct the matrix of the morphism $\mathscr{M}$. Then, we detail the cases were the transitions are defined by non-linear systems, i.e., when $d \leq 2$. See the condition at line 7. Then, we define T as a polynomial of degree $dr - r$ in its canonical form, i.e., with parameterized coefficients. See `Template_Canonical_Form`$(n, d, r, dr-r)$, at line 7. Here, `Template_Canonical_Form`$(n, d, r, dr-r)$ returns the lists of unknown coefficient that we denoted by $\lambda_0, \ldots, \lambda_s$ in Sect. 5.2. Note that s depends only on the values of n, d, and r. One can now call the function `Matrix_L`(r, dr, T) that construct the matrix L as shown in Sect. 5.2.

Next, we apply our decision procedure `Reduce_Rank_Assigning_Values`$(M - L)$ to assign values to the coefficients of T in such a way that $Ker(M-L) \neq 0$. See line 10. As we saw in the previous section, $Ker(M-L) \neq 0$ is equivalent to having $Rank(M - L) < Dim(\mathbb{R}_r[X_1, \ldots, X_n])$. In other words, it is the same as having $M - L$ with rank strictly less than the dimension $v(r)$ of $\mathbb{R}_r[X_1, \ldots, X_n]$. We then reduce the rank of $M - L$ by assigning values of the parameters in L. This function applies the decision procedure detailed in Sect. 5.4, Step 2. Next, we determine whether the matrix obtained, $\overline{M - L}$, has a trivial kernel by first computing its rank and then checking if **Rank**$(\overline{M - L}) <$ **Dim**$(R_r[X_1, .., X_n])$ holds, at line 11. We can now apply our main Theorem 5:

- If $\overline{M - L}$ has a trivial kernel, we know there is no T-scale invariants of degree less than r and we can increase the degree r of the desired invariants until Theorem 5, or Corollary 3, applies, or until stronger hypotheses occur, e.g. if all $v(r) \times v(r)$ sub-determinants are null. Note, at line 12, the call to `return Ideal_Loop_Inv_Gen`$(r+1, P_1, \ldots, P_n, X_1, \ldots, X_n)$. If there is no ideal for non-trivial invariants for a value r_i then we conclude that there is no ideal of non-trivial invariants for all degrees $k \leq r_i$. This can also be used to guide other constraint-based techniques, since checking for invariance with a template of degree less or equal to r_i will not be necessary.
- Otherwise, Theorem 5, or Corollary 3, guarantees the existence and computation of T-invariants. Finally, the function `Nullspace_Basis`$(\overline{M - L})$ outputs the basis of the nullspace of the matrix $\overline{M - L}$, in order to construct non-trivial invariants. See line 15.

For basis computations, we use well-known state-of-the-art algorithms, for example those that `Sage` provides. These algorithms calculate the eigenvalues and associated eigenspaces of $\overline{M - L}$ when it is a square matrix. When $\overline{M - L}$ is a rectangular matrix, we can use its *singular value decomposition* (SVD). A SVD of $\overline{M - L}$ provides an explicit representation of its rank and kernel by computing unitary matrices U and V and a regular diagonal matrix S such that $\overline{M - L} = USV$. We compute the SVD of a $v(r + d - 1) \times v(r)$ matrix $\overline{M}$ in two steps. First, we reduce it to a bi-diagonal matrix, with a cost of $O(v(r)^2 v(r + d - 1))$ flops. The second step relies on an iterative method, as is also the case for other algorithms that compute eigenvalues. In practice, however, it suffices to compute the SVD up to a certain precision, i.e. up to a machine epsilon. In this case, the second step takes $O(v(r))$ iterations, each using $O(v(r))$ flops. So, the overall cost is $O(v(r)^2 v(r + d - 1))$ flops. For the encoding of the algorithm we could rewrite Corollary 3 as follows.

Corollary 4 *Let $(M - L)' = USV$ be the singular value decomposition of matrix $(M - L)'$ described just above. There will be a non-trivial T-invariant for any given initial condition if and only if the number of non-zero elements in matrix S is less than $v(r) - 2$, where $v(r)$ is the dimension of $\mathbb{R}_r[x_1, \ldots, x_n]$. Moreover, the orthonormal basis for the nullspace obtained from the decomposition directly gives an ideal for non-linear invariants.* □

Algorithm 1: **Ideal_Loop_Inv_Gen(**$r, P_1, \ldots, P_n, X_1, \ldots, X_n$**)**

/*Finding degree bounds for discrete transitions.*/;
Data: r is the candidate degree for the set of basis invariants elements we are looking for, $P_1, ..P_n$ the n polynomials given by the considered loop, and $X_1, ..X_n \in V$
Result: $Ideal\_Inv$, a basis of ideal of invariants.
begin
    int d;
    Template T;
    Matrix M, L;
    $d \longleftarrow$ **Max_degree(**$\{P_1, \ldots, P_n\}$**)**;
    /*d is the maximal degree of P_i's*/;
    $M \longleftarrow$ **Matrix_D(**$r, dr, P_1, \ldots, P_n$**)**;
    if $d >= 2$ **then**
        $T \longleftarrow$ **Template_Canonical_Form(**$n, d, r, dr - r$**)**;
        $L \longleftarrow$ **Matrix_L(**r, dr, T**)**;
        $\overline{M - L} \longleftarrow$ **Reduce_Rank_Assigning_Values(**$M - L$**)**;
        if **Rank(**$\overline{M - L}$**)** $>=$ **Dim(**$R_r[X_1, .., X_n]$**) then**
            return Ideal_Loop_Inv_Gen($r + 1, P_1, \ldots, P_n, X_1, \ldots, X_n$**)**;
            /*We need to increase the degree r of candidates invariants.*/;
        else
            return Nullspace_Basis($\overline{M - L}$**)**;
            /*There exists an ideal of invariants that we can compute*/;
    else
        ... /*We refer to our previous work for constant scaling.*/;
end

Remark 4 It is important to emphasize that eigenvectors or nullspace of $\overline{M - L}$ are computed after the parameters of L_T have been assigned. When the discrete transition system has several variables and none or few parameters, which correspond to practical cases, $\overline{M - L}$ will be over the reals and there will be no need to use the symbolic version of these algorithms. □

7. Invariant generation for discrete transitions and fractional systems

We now want to deal with transition systems ρ_τ of the following type

$$\begin{bmatrix} X_1' = \frac{P_1(X_1,\ldots,X_n)}{Q_1(X_1,\ldots,X_n)} \\ \vdots \\ X_n' = \frac{P_n(X_1,\ldots,X_n)}{Q_n(X_1,\ldots,X_n)}, \end{bmatrix}.$$

where the P_i's and Q_i's belong to $\mathbb{R}[X_1, .., X_n]$ and each P_i is relatively prime to the corresponding Q_i. In this case, one needs to relax the consecution conditions to fractional-scale as soon as fractions appear in the transition relation.

Theorem 8 *A polynomial Q in $\mathbb{R}[X_1, .., X_n]$ is a F-scale invariant for fractional discrete scale consecution with a* parametric fractional $F \in \mathbb{R}(X_1, .., X_n)$ *for τ if and only if*

$$Q\left(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}\right) = FQ.$$

□

Let d be the maximal degree of the P_i's and Q_i's, and let Π be the least common multiple of the Q_i's. Now let $U = {X_1}^{i_1}..{X_n}^{i_n}$ be a monomial of degree less than r, *i.e.*, $i_1 + .. + i_n < r$. Then,

$$\Pi^r U(P_1/Q_1, \dots, P_n/Q_n) = \Pi^r (P_1/Q_1)^{i_1} \dots (P_n/Q_n)^{i_n}.$$

But as $Q_j^{i_j}$ divides Π^{i_j}, for all j, we see that $Q_1^{i_1} \dots Q_n^{i_n}$ divides $\Pi^{i_1+\dots+i_r}$, which divides Π^r. We conclude that $\Pi^r Q(P_1/Q_1, \dots, P_n/Q_n)$ is a polynomial for every Q in $\mathbb{R}_r[X_1, .., X_n]$.

Now suppose that $F = T/S$, with T relatively prime to S, satisfies the equality of the previous theorem. Suppose, further, that we are looking for bases for invariants Q of degree r. Then, multiplying by Π^r we get

$$\Pi^r Q(P_1/Q_1, \dots, P_n/Q_n) = (\Pi^r TQ)/S.$$

As we have no a priory information on Q, in most cases Q will be relatively prime to S. In this situation we see that S divides Π^r. So, let F be of the form T/Π^r, and note that we argued that this constraint is weak.

Now let $\mathscr{M}$ be the morphism

$$\mathscr{M} : \begin{cases} \mathbb{R}_r[X_1, \dots, X_n] \to \mathbb{R}_{nrd}[X_1, \dots, X_n] \\ \qquad\qquad Q \mapsto \Pi^r Q\left(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}\right). \end{cases}$$

Let M be its matrix representation in the canonical basis, let T be a polynomial in $\mathbb{R}_{nrd-r}[X_1, .., X_n]$, and let $\mathscr{L}$ denote the vector space morphism

$$\mathscr{L} : \begin{cases} \mathbb{R}_r[X_1, \dots, X_n] \to \mathbb{R}_{nrd}[X_1, \dots, X_n] \\ \qquad\qquad Q \mapsto TQ. \end{cases}$$

Also, let L be its matrix representation in the canonical basis. As stated in the following theorem, our problem is equivalent to finding a L such that $M - L$ has a non-trivial kernel.

Theorem 9 *Consider M and L as described above. Then, there exist F-scale invariants, where F is of the form T/Π^r, if and only if there exists a matrix L such that $Ker(M - L) \neq \emptyset$. In this situation, any vector in the kernel of $M - L$ will give rise to a F-scale discrete invariant.* □

This is similar to Theorems 6 and 7. For the initiation step, we have a hyperplane in $\mathbb{R}_r[X_1, \dots, X_n]$. In order for the transition system to make sense, the n-tuple of initial values must not be a root of any of the Q_i's, and similarly for further iterations as long as the loop is applied. In this way, they will not cancel Π^r. We have the following result.

Theorem 10 *We have a non-trivial invariant if and only if there exists a matrix L such that the intersection of the kernel of $M - L$ and the hyperplane given by the initial values is not zero. The invariants will correspond to vectors in the intersection.* □

We also have the following important corollary.

Corollary 5 *We will have a non-trivial invariant for any non-trivial initial value if and only if there exists a matrix L such that the dimension of $Ker(M - L)$ is at least 2.* □

Example 8 Consider the system $\rho_\tau \equiv \begin{bmatrix} x_1' = \frac{x_2}{(x_1+x_2)} \\ x_2' = \frac{x_1}{(x_1+2x_2)} \end{bmatrix}$. We are looking for F-scale invariant polynomials of degree 2. The least common multiple of $(x_1 + x_2)$ and $(x_1 + 2x_2)$ is their product, so that $\mathscr{M}$ is given by:

$$Q \in \mathbb{R}_2[x_1, x_2] \mapsto \left[[(x_1 + x_2)(x_1 + 2x_2)]^2 Q\left(\frac{x_1}{(x_1 + x_2)}, \frac{x_2}{(x_1 + 2x_2)}\right)\right].$$

As both $\frac{x_2}{(x_1+x_2)}$ and $\frac{x_1}{(x_1+2x_2)}$ have degree zero,

$$[(x_1 + x_2)(x_1 + 2x_2)]^2 Q\left(\frac{x_2}{(x_1 + x_2)}, \frac{x_1}{(x_1 + 2x_2)}\right)$$

will be a linear combination of degree 4, if it is non-null.

Hence, $\mathscr{M}$ has values in $Vect(x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4)$. With T and Q in $\mathbb{R}_2[x_1, x_2]$ we verify that

$$[(x_1 + x_2)(x_1 + 2x_2)]^2 Q\left(\frac{x_2}{(x_1 + x_2)}, \frac{x_1}{(x_1 + 2x_2)}\right) = TQ.$$

As the left member is in $Vect(x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4)$, T must be of the form $\lambda_0\, x_1^2 + \lambda_1\, x_1x_2 + \lambda_2\, x_2^2$, and Q must be of the form $a_0x_1^2 + a_1x_1x_2 + a_3x_2^2$. We see that we can take Q in $Vect(x_1^2, x_1x_2, x_2^2)$, and similarly for T. Then both $\mathscr{M}, \mathscr{L} : Q \mapsto TQ$ are morphisms from $Vect(x_1^2, x_1x_2, x_2^2)$ into $Vect(x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4)$. In the corresponding canonical basis, the matrix $M - L$ is

$$M - L = \begin{pmatrix} -\lambda_0 & 0 & 1 \\ -\lambda_1 & 1-\lambda_0 & 2 \\ 1-\lambda_2 & 3-\lambda_1 & 1-\lambda_0 \\ 4 & 2-\lambda_2 & -\lambda_1 \\ 4 & 0 & -\lambda_2 \end{pmatrix}.$$

Taking $\lambda_0 = 1$, $\lambda_1 = 3$ and $\lambda_2 = 2$, the second column cancels out and the kernel will be equal to $Vect(0, 1, 0)$. Now, Corollary 5 applies to $M - L$, and we obtain: $T(x_1, x_2)/Q(x_{1,2}) = 1/((x_1 + x_2)(x_1 + 2x_2))^2$ and we have the nullspace `[(0, 1, 0)]` and the basis of scale invariant $\{x_1x_2\}$. It was clear from the beginning that the corresponding polynomial x_1x_2 is $\frac{1}{[(x_1+x_2)(x_1+2x_2)]^2}$-scale invariant. In particular, it is an invariant for the initial values $(0, 1)$. Moreover, it clearly never cancels $x_1 + x_2$ and $x_1 + 2x_2$, because they are of the form $(a, 0)$ or $(0, b)$ with a and b strictly positive. $\square$

8. Branching conditions and nested loops

We have generated bases of vector spaces describing invariants for transition systems. A global invariant would be any invariant which is in the intersection of these vector spaces. In this way, we avoid the definition of a single isomorphism for the whole transition system. Instead, we generate the basis for each separate consecution condition. To compute a basis of global invariants, we could use the following theorem. It suggests to *multiply* all the elements of each computed basis. By so doing, we also avoid the heavy computation of ideal intersections.

Theorem 11 *Let $I = \{I_1, \ldots, I_k\}$ a set of ideals in $\mathbb{R}[X_1, \ldots, X_n]$ such that $I_j = (f^{(j)}{}_1, \ldots, f^{(j)}_{n_j})$ for $j \in [1, k]$. Let $\otimes(I_1, \ldots, I_k) = \{\delta_1, \ldots, \delta_{n_1n_2\ldots n_k}\}$ be such that all elements δ_i in $\otimes(I_1, \ldots, I_k)$ are formed by the product of one element from each ideal in I. Assume that all I_j's are ideals for invariants for a loop at location l_j, described by a transition τ_j. If all l_j describe the same location or program point l, then $\otimes(I_1, \ldots, I_k)$ is an ideal of non-trivial non-linear invariants for the entire loop located at l.* $\square$

Note that when we have several transitions looping at the same point, we can obtain an encoding of possible execution paths of a loop containing conditional statements.

This approach is a sound, but not complete, way of computing ideals for global invariants, and it also has a low computational time complexity. In order to take into account initial conditions we intersect these vector spaces with the initial semi-hyperplanes deduced from the isomorphism associated with initial requirements. Next, we show how our method deals with the conditional statements inside loops. Let's consider the following type of loop `while(B_1){ [I_1;] if(B_2){[I_2;] } else{ [I_3;] } [I_4;]`, where each I_i represent a block of multivariate fractional instructions. First we represent the loop with the following two transitions $\tau_1 = \langle l_i, l_i, (\mathcal{B}_1 \wedge \mathcal{B}_2), \rho_{\tau_1}\rangle$ and $\tau_2 = \langle l_i, l_i, (\mathcal{B}_1 \wedge \neg\mathcal{B}_2), \rho_{\tau_2}\rangle$, where: $\rho_{\tau_1} \equiv [x_1' = F_{1,[I_1;\, I_2;\, I_4;\,]_\circ}(x_1, \ldots x_n), \ldots, x_n' = F_{n,[I_1;\, I_2;\, I_4;\,]_\circ}(x_1, \ldots, x_n)]$ and $\rho_{\tau_2} \equiv [x_1' = F_{1,[I_1;\, I_3;\, I_4;\,]_\circ}(x_1, \ldots x_n) \ldots, x_n' = F_{n,[I_1;\, I_3;\, I_4;\,]_\circ}(x_1, \ldots, x_n)]$, with $[;\ ;\ ]_\circ$ denoting our operator, based on separation rewriting rules, used to compose blocks of instructions. We first independently generate the ideals of invariants $\xi_1 = (\mu_1, \ldots, \mu_n)$ and $\xi_2 = (\kappa_1, \ldots, \kappa_p)$ for the respective transitions τ_1 and τ_2. Any element $\mu_i \in \xi_1$ refers to an inductive invariant $\mu_i(X_1, \ldots, X_n) = 0$ corresponding to the *partial loops* described by transition τ_1. Similarly, any $\kappa_i \in \xi_2$ refers to an inductive invariant $\kappa_i(X_1, \ldots, X_n) = 0$ for the loop described by transition τ_2. Then we can take $\mu_i(X_1, \ldots, X_n) * \kappa_i(X_1, \ldots, X_n) = 0$ as global loop invariant, since these invariants will remain true in any sequence of transitions during the execution of the loop. We deal with loop conditions using the same methods that we proposed to handle initiation conditions. We know, for instance, that if our Corollary 3 holds, then there exist invariants for any (semi-)hyperplane that could be induced by the loop conditions. We illustrate this point in Fig. 1. Let $(P_i(x_1, .., x_n) < 0)$ be semi-algebraic loop conditions at location l and let Q be an inductive invariant for $\mathcal{D}(l)$. Thus $(P_i(x_1, .., x_n) - Q(x_1, .., x_n) < 0)$ is also an inductive invariant. Then, we can build an operator, similar to the one introduced in Theorem 11, to generate, in a different way, ideals of non-trivial invariants at a state l with semi-algebraic loop conditions. If a loop condition has the form $C_i(x_1, .., x_n) = 0$ we could then associate it directly to polynomial systems induced by the transition relations.

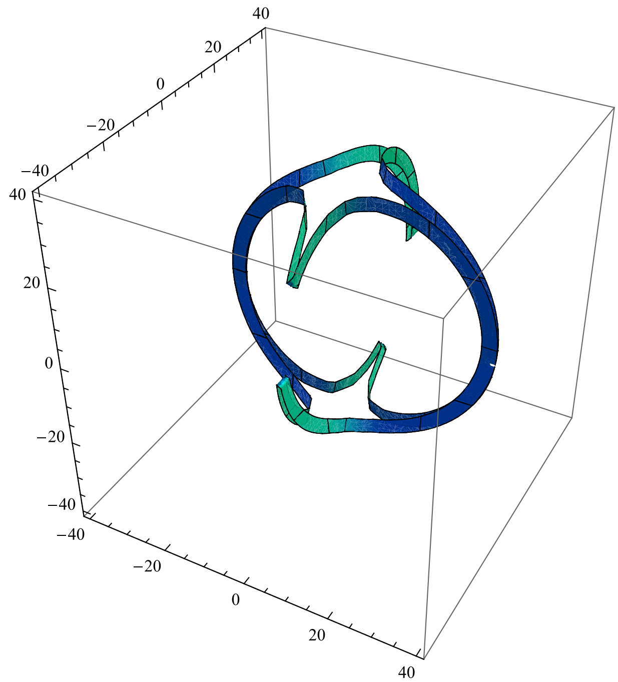

Fig. 1. Intersection between the conditional loop: $800 < (x-5)^2+(y-5)^2+(y_0-5)^2 < 1000$ and the invariant $y_0(1-y_0)x^2+xy-x+y^2-2y+1 = 0$ from the invariant ideal $(\{x^2, xy-x, y^2-2y+1\})$ obtained for Example 7

Example 9 Consider the following loop.

```
int u_0; //initialization
((M > 0)&&(Z = 1)&&(U = u_0)...)
       ...
  While ((X>=1) || (Z>=z_0)){
       If(Y > M){
            X = Y / (X + Y);
            Y = X / (X + 2 * Y);}
       Else{
           Z = Z * (U + 1);
           U = U^2; }
  }
```

We first generate an invariant for the loop corresponding to the first conditional `if`. Using Fractional-Scaling we obtain the basis of scale invariant $\{\{xy\}\}$. See Example 8 for more details. Then, we obtain the basis of invariants $\{u_0z^2 - u_0^2z^2 + zu + u^2 - z - 2u + 1, \ldots\}$ corresponding to the other alternative transition τ_2 of the loop, namely, the `Else` clause. Now we return the global invariants:
$\{xyu_0z^2 - u_0^2z^2 + xyzu + xyu^2 - xyz - 2xyu + xy, \ldots\}$ So, $xyu_0z^2 - u_0^2z^2 + xyzu + xyu^2 - xyz - 2xyu + xy = 0$ is one typical invariant that can be generated. Once again, here there are no need for Gröbner basis computation and the complexity of the described steps remains polynomial. □

Example 9 illustrate our method for the case where the loop contains two conditional statements. In the presence of nested loops, our method generates ideals for invariants for each inner-loop and then generates a global invariant.

9. Experiments

The third column in Table 1a summarizes the type of linear algebraic problems associated with each kind of consecution approximation, listed in the second column, and with the semantic of the program instructions appearing in the first column. The last column in Table 1a gives some existential results which, we note, can also be used by other constraint-based approaches or reachability analysis methods. We have also used it to obtain some experimental results that attest to the effectiveness and scope of our methods considering the computation

Table 1. Examples and experimental results

(a) Linear algebraic problems and consecution approximations

Prog. Loop	Aprox.Consec.	Linear Algebra	Existence Cond.
Affine/lin. inst.	Strong Scaling	Nullspaces	$Dim\_Ker(M_D) \geq 2$ for any init. cond., and $Ker(M_D) \neq \emptyset$ otherwise.
Affine/lin. inst.	Lambda Scaling	Eigenspaces	$Dim\_Eigen(M_D) \geq 2$ for any init. cond., and $Eigen(M_D) \neq \emptyset$ otherwise.
Algebraic/poly. inst.	Polynomial Scaling	Nullspaces	$Ker(M_D - L_T) \geq 2$ for any init. cond., and $Ker(M_D - L_T) \neq \emptyset$ otherwise.
Fractional inst.	Fractional Scaling	Nullspaces	$Dim\_Ker(M_\Pi - L_T) \geq 2$ for any init. cond., and $Ker(M_\Pi - L_T) \neq \emptyset$ otherwise.

(b) Experimental results: computation of nullspaces and eigenspaces

Loop prog.	Var.	Par.	Scaling	Basis inv.	CPU (s)
1 - $\begin{bmatrix} x_1' = 2x_1 + x_2 + 1 \\ x_2' = 3x_2 + 4 \end{bmatrix}$	$\{x_1, x_2\}$		$\lambda \in \{9, 6, 4, 3, 2, 1\}$	$\{\{x_2^2 + 4x_2 + 4\};\ ...\}$	0.39
2 - $[s' = s + i;\ j' = j - 1;\ i' = i;\ j_0' = j_0]$	$\{s, i, j, j_0\}$		$\lambda \in \{1\}$	$\{\{ji + s, i^2, ij_0, i, j_0, 1\}\}$	1.27
3 - $\begin{bmatrix} x_1' = ax_1 + bx_2 + c \\ x_2' = dx_1 + ex_2 + f \end{bmatrix}$	$\{x_1, x_2\}$	$\{a, b, c, d, e, f\}$	$\lambda \in \{0, d^2 + \frac{1}{2}de + \frac{1}{2}e^2 \pm \frac{1}{2}sqrt(8d^3e + d^2e^2 + 6de^3 + e^4), ...\}$	$\{\{x_1 - 1\};\ ...\}$	1.35
4 - $(\tau_1) : [r' = r + 1;\ w' = 0;\ k' = k - c_1;\ c_1 = c_1]$	$\{r, w, k, c_1\}$		$\lambda \in \{0, 1\}$	$\{\{rw, w^2, wk, wc_1, w, k + c_1\};\ \{c_1^2, c_1, 1\}\}$	0.37
5 - $(\tau_2) : [r' = 0;\ w' = w + 1;\ k' = k - c_2;\ c_2 = c_2]$	$\{r, w, k, c_2\}$		$\lambda \in \{0, 1\}$	$\{\{r^2, rw, rk, rc_2, r\};\ \{wc_2 + k\}\}$	0.4
6 - $(\tau_3) : [r' = r - 1;\ w' = 0;\ k' = k + c_1;\ c_1 = c_1]$	$\{r, w, k, c_1\}$		$\lambda \in \{0, 1\}$	$\{\{rw, w^2, wk, wc_1, w\};\ \{rc_1 + k, c_1^2, c_1, 1\}\}$.	0.39
7 - $(\tau_4) : [r' = 0;\ w' = w - 1;\ k' = k + c_2;\ c_2 = c_2]$	$\{r, w, k, c_2\}$		$\lambda \in \{0, 1\}$	$\{\{r^2, rw, rk, rc_2, r\};\ \{c_2^2, w, c_2, 1\}\}$.	0.43
8 - $\begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix}$	$\{x, y\}$		$T(x, y) = y^2 + 2y + 1$	$\{\{x^2, xy - x, y^2 - 2y + 1\}\}$	0.4
9 - $\begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix}$	$\{x, y\}$		$T(x, y) = y^2 + y$	$\{\{xy, y^2 - y\}\}$	0.41
10 - $\begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix}$	$\{x, y\}$		$T(x, y) = y^2$	$\{\{y^2\}\}$	0.41
11 - $\begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix}$	$\{x, y\}$		$T(x, y) = y + 1$	$\{\{x, y - 1\}\}$	0.47
12 - $\begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix}$	$\{x, y\}$		$T(x, y) = y$	$\{\{y\}\}$	0.37
13 - $\begin{bmatrix} x_1' = \frac{x_2}{(x_1 + x_2)} \\ x_2' = \frac{x_1}{(x_1 + 2x_2)} \end{bmatrix}$	$\{x_1, x_2\}$		$T(x_1, x_2) = x_1^2 + 3x_1x_2 + 2x_2^2$	$\{\{x_1x_2\}\}$	0.37
14 - $\begin{bmatrix} x' = ax^2 + dx \\ y' = axy + dy \end{bmatrix}$	$\{x, y\}$	$\{a, d\}$	$T(x, y) = a^2x^2 + 2adx + d^2$	$\{\{x^2, xy, y^2\}\}$	0.48
15 - $\begin{bmatrix} x' = ax^2 + dx \\ y' = axy + iy^2 + dy \end{bmatrix}$	$\{x, y\}$	$\{a, d, i\}$	$T(x, y) = a^2x^2 + 2adx + d^2$	$\{\{x^2\}\}$	0.42
16 - $\begin{bmatrix} x' = ax^2 + bxy + dx \\ y' = axy + by^2 + dy \end{bmatrix}$	$\{x, y\}$	$\{a, b, d\}$	$T(x, y) = ax + by + d$	$\{\{x, y\}\}$	0.43
17 - $\begin{bmatrix} x' = bxy + cy^2 + x \\ y' = y \end{bmatrix}$	$\{x, y\}$	$\{b, c, d\}$	$T(x, y) = 1$	$\{\{y^2, y, 1\}\}$	0.45
18 - $[x' = bxy + cy^2 + dx + ey + f;\ y' = bx^2 + cxy + dx + ey + f]$	$\{x, y\}$	$\{b, c, d, e, f\}$	$T(x, y) = -bx - cy$	$\{\{x - y\}\}$	0.54
19 - $[x' = ax^2 + bxy;\ y' = gx^2 + hxy + iy^2 + ky]$	$\{x, y\}$	$\{a, b, g, h, i, k\}$	$T(x, y) = ax + by$	$\{\{x\}\}$	0.47
20 - $\begin{bmatrix} x' = ax^2 + bxy \\ y' = axy + by^2 \end{bmatrix}$	$\{x, y\}$	$\{a, b\}$	$T(x, y) = ax + by$	$\{\{x, y\}\}$	0.46
21 - $\begin{bmatrix} x' = ax^2 + cy^2 + f \\ y' = ax^2 + cy^2 + l \end{bmatrix}$	$\{x, y\}$	$\{a, c, f, l\}$	$T(x, y) = 0$	$\{\{x^2, 2 * ((f * g...\};\ ...\}$	1.84

of eigenspaces or nullspaces. By using efficient mathematical packages, e.g. `Sage` [SJ05], `Maple`, `Mathematica`, `Lapack` or `Eispack`, one can obtain the eigenvalues as closed-form algebraic expressions. That is, algebraic numbers which comprise the solutions of an algebraic equation in terms of its coefficients, relying only on $+,-,*,/$, and the extraction of roots. Also, eigenvalues are already obtained as algebraic numbers in practice, for large well known classes of matrices, such as when $n < 5$, and when the matrix is 5×5 block triangular, among others. From Table 1a, we note that the computation of nullspaces or eigenspaces remain the main computational steps in our approaches. Table Sec-experimentb lists some of these experimental results focusing on the type of systems, scaling and basis of invariants that one could expect using our approaches. The first column refers to the specific problem of the experiment. The second column provides the numbers of variables. The third column gives the parameters used to represent a program class. The column `Scaling` shows the approximation of the consecution conditions. The column `Basis Inv.` presents the types of basis of invariants that one can generate. The last column refers to the cpu time required to compute those nullspaces or eigenspaces that turn out as vector spaces of invariants. We can see that our methods efficiently handle a large number of non-linear examples treated elsewhere in the literature. The experiment 2 is from [SSM04b] and the experiments 4–7 relate to the loop transitions of the generalized readers-writers case studies from [SSM04b]. Experiments 8–20, listed in Table 1b, involve non-linear systems most of which can be shown to be beyond the limits of other recent approaches. The experiments 8–11, expose the types of basis of invariants that one can generate considering four different polynomial scalings. More important experiment 3 and experiments 13–20 refer to generic programs, i.e., large classes of programs and the associated generic basis of invariants that we provide. We use parameters to represent these classes of programs and we generate generic basis of invariants. Those parameters could also be use to abstract away some variables of a larger program. The experiment 12 involves a fractional system. We used the very complete `Sage` [SJ05] algebraic framework with interfaces written in `Python` so as to be able to access useful mathematical packages. Although the main contribution of this work is theoretical—we present theorems that could well be used with, or complement, other existing formal methods—the computation of these specific nullspaces or eigenspaces was conducted and depended on the `Sage`'s on-line servers available. Those results show the strength of our approach for generating non-linear invariants for non-linear systems.

10. Discussion

The notions of Gröbner bases and their computations, together with the *ideal membership problem* are central to most recent approaches to program verification and static analysis [SSM04b, BBGL00, RCK07a, BLS96, CXYZ07, Kov08, KJ06, Cou05, MOS02, RCK07b, GT08, PC08]. In order to better understand the difficulties they incur, we first need some details on Gröbner basis and the ideal membership problem. Consider a multivariate polynomial, $Q = \sum_{i_1,\ldots,i_n} a_{i_1,\ldots,i_n} {x_1}^{i_1} \ldots {x_n}^{i_n}$, where the coefficients $a_{i_1,\ldots,i_n}$ are in a field K. How do we know if it is in an ideal I of $K[X_1, \ldots, X_n]$? This is known as the *Ideal membership problem*. To handle it we can use a Gröbner basis $G = \{g_1, \ldots, g_s\}$ for I. There are algorithms that compute such bases as long as we know a finite generating basis for I [Buc96, Fau99]. Then, we can compute the normal form of Q for I using the basis G. Denote the normal form by $NF_G(Q)$. We note that the use of a Gröbner basis guarantees the confluence and termination of those reductions. In general, we have $NF_G(Q) = \sum_{i_1,\ldots,i_n} f(a)_{i_1,\ldots,i_n} {x_1}^{i_1} \ldots {x_n}^{i_n}$, where $f(a)_{i_1,\ldots,i_n}$ is a combinations of the coefficients $a_{i_1,\ldots,i_n}$. Then the statement $(Q \in I)$ is equivalent to the assertion $(NF_G(Q) = 0)$, that is, all the coefficients $f(a)_{i_1,\ldots,i_n}$ are null.

Returning to the mentioned approaches for program verification and static analysis, the loop instructions are considered in order to form varieties and to build associated algebraic assertions and the ideal I. Then, these techniques compute a Gröbner basis G for I. Next, they postulate a template polynomial Q, i.e., a polynomial with unknown coefficients, as a *candidate invariant*. As we have seen just above, Q is an invariant if it belongs to the ideal I or, in other words, if $(NF_G(Q) = 0)$. So, the next step in these techniques, is to obtain the reduction $NF_G(Q)$. An important obstacle faced at this point is that all known algorithms for computing Gröbner basis and for the construction of the normal form reduction $NF_G(Q)$ are of doubly exponential time complexity. Having the normal form $NF_G(Q)$, they generate the set of *candidate invariant constraints* in the form of the system of equations $(NF_G(Q) = 0)$, and attempt to solve it directly. Moreover, as we have seen in Sect. 4.3, as soon as the loop contains a non-linear instruction, the candidate invariant constraints results in a non-linear system of equation, which makes its resolution all but unfeasible. Further, there are no conditions over the degree of their candidate invariants that would guarantee the non-triviality of the resulting invariant, when it can be computed.

In terms of performance and efficiency, we succeeded in reducing the non-linear loop invariant generation problem to a linear algebraic problem, i.e., to the computation of eigenspaces of specific morphisms. Our techniques have fewer computational steps: we first compute some specific matrices and then we compute their nullspaces. Each of these steps remains of polynomial time complexity. Further, our approaches do not simply generate an invariant at a time. Instead, we generate an ideal of invariants which is a large—infinite—structure. We also handle fractional systems and our algorithm incorporates a strategy to find degree bounds which allow for the automatic generation of ideals of non-trivial invariants. Moreover, as one of the main results, we provide very general sufficient conditions allowing for the existence and computation of such invariant ideals. Note that these conditions could be directly used by any other invariant generation methods.

As a more applied motivation, our techniques can be made to bear on new domains that require the computation of complex invariants. Along these lines, some recent work on security [RM11c, RM09, RM11b, RM11a], showed how such invariants play a central role in static analysis of malwares, e.g., viruses, and how they can be used to build new invariant-based intrusion detection systems. Invariants generated over malware codes are strong semantic aware signatures that can be used to analyze and identify intrusions caused by such malicious code. These new approaches could form parts of intrusion detection systems where an alarm is a proof of abnormal behavior caused by the violation of a precomputed invariant induced by the intrusion. We note that binary code gives rise to non-linear arithmetic and the methods described here allow, as we have shown, for the generations of complex and precise invariants in such cases. We stress that the more the complex the invariant is, the harder it will be to morph the corresponding signatures in an automatic way.

11. Conclusions

Our primary goal and motivation were to provide invariant generation methods for static analysis that could serve as a basis for automatic program verification.

We have shown that the preconditions for discrete transitions can be viewed as morphisms over a vector space of bounded degree polynomials. These morphisms, in turn, could be suitably represented by matrices. By doing so, we succeeded in reducing the non-linear loop invariant generation problem to linear algebraic problems or, more precisely, to the computation of eigenspaces of these morphisms. We also treated fractional systems and our algorithms incorporate a strategy to find degree bounds for candidate invariants, thus allowing for the automatic generation of non-trivial invariants.

These techniques lead to algorithms of much lower time complexity than other modern approaches. The latter incur in computations which are of a doubly exponential time complexity while, by contrast, our techniques induce algorithms of polynomial time complexity.

Further, our methods do not generate a single invariant at a time. Instead, we generate non-linear invariant ideals, which are infinite structures, giving rise to families of non-trivial invariants. As another important main result, we provided very general sufficient conditions that can guarantee the existence of such invariant ideals.

We also noted that our techniques could be combined with other formal verification methods and their associated tools. A case in point are formal methods that treat logics with uninterpreted functions [GT06], which can handle function calls and operating system calls.

Acknowledgements

We would like to thank the anonymous reviewers for their detailed comments.

A. Appendix

Proof of Theorem 1 If $Q(X'_1, .., X'_n) - \lambda Q(X_1, .., X_n)$ belongs to the ideal I generated by the family $(X'_1 - L_1, \dots, X'_n - L_n)$, then there exists a family $(A_1, \dots, A_n)$ of polynomials in $\mathbb{R}[X'_1, \dots, X'_n, X_1, \dots, X_n]$ such that $Q(X'_1, \dots, X'_n) - \lambda Q(X_1, \dots, X_n) = (X'_1 - L_1)A_1 + \cdots + (X'_n - L_n)A_n$. Letting $X'_i = L_i$, we obtain $Q(L_1(X_1, \dots, X_n), \dots,$
$L_n(X_1, \dots, X_n)) = \lambda Q(X_1, \dots, X_n)$. Conversely suppose

$Q(L_1(X_1, .., X_n), .., L_n(X_1, .., X_n)) = \lambda Q(X_1, .., X_n)$. Then as $Q(X'_1, \dots, X'_n)$ is equal to $Q(L_1, .., L_n)$ modulo the ideal I, we get $Q(X'_1, .., X'_n) = \lambda Q(X_1, \dots, X_n)$ modulo I. □

Proof of Theorem 2 Let Q be a polynomial in $\mathbb{R}_r[X_1, .., X_n]$. We have the following sequence of deduction: $(Q(L_1(X_1, \dots, X_n), ..,$
$L_n(X_1, \dots, X_n)) = \lambda Q(X_1, \dots, X_n)) \Leftrightarrow (\mathscr{M}(Q) = \lambda Q) \Leftrightarrow (\mathscr{M}(Q) = \lambda\, Id(Q)) \Leftrightarrow ((\mathscr{M} - \lambda\, Id)(Q) = 0_{\mathbb{R}[X_1,...,X_n]}) \Leftrightarrow (Q \in Ker(M - \lambda\, I))$. Using the definition of an invariant and Theorem 1, we can see that Q will be a λ-scale invariant if and only if it belongs to the eigenspace corresponding to λ. □

Proof of Corollary 1 Suppose M is block triangular with blocks 4×4 or less, then its characteristic polynomial will a product of polynomials of degree less than four, whose roots can be calculated by the Lagrange resolvent method [Lan02]. For the second assertion, we already know that 1 is an eigenvalue. Suppose that the corresponding eigenspace is of dimension exactly one. Then the only vectors in that space are the constant polynomials. If it is of dimension two or more, then we get non-trivial polynomials in the eigenspace. □

Proof of Theorem 3 We first consider Theorem 2. The initiation step defines on $\mathbb{R}_r[x_1, \dots, x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1, \dots, u_n)$. Hence, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1, .., X_n]$ given by the kernel of I_u, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] \mid Q(u_1, \dots, u_n) = 0\}$. If we add initial conditions of the form $(x_1(0) = u_1, \dots, x_n(0) = u_n)$, we are looking for a λ-scale invariant in $\mathbb{R}_r[x_1, \dots, x_n]$ that belongs to the hyperplane $P(u_1, \dots, u_n) = 0$, *i.e.*, we are looking for Q in $ker(M - \lambda\, I) \cap \{P \mid P(u_1, \dots, u_n) = 0\}$. □

Proof of Corollary 2 We take each direction, in turn. [$(\Rightarrow)$] There is a λ-scale invariant for any initial value. Then the corresponding eigenspace has dimension at least 2. Indeed, if the space was of dimension only 1 (which is at least necessary to have λ-invariants), than taking any nonzero vector Q in the eigenspace (i.e. a λ-invariant), Q should lie in any hyperplane of initial values. That is for every n-tuple $(u_1, \dots, u_n)$ one would have $Q(u_1, \dots, u_n) = 0$, hence $Q = 0$, which is absurd. [$(\Leftarrow)$] Any eigenspace of M with dimension at least 2 will intersect any space, in particular any hyperplane, given by any initial constraints. As any hyperplane is of co-dimension one in $\mathbb{R}_r[X_1, \dots, X_n]$, it must have a nonzero intersection with any subspace of dimension strictly greater than one. This establishes the result. □

Proof of Theorem 4 [$(\Rightarrow)$] If $Q(X_1', .., X_n') - TQ(X_1, .., X_n)$ belongs to the ideal I generated by the family $(X_1' - P_1, \dots, X_n' - P_n)$, then there exists a family $(A_1, \dots, A_n)$ of polynomials in $\mathbb{R}[X_1', .., X_n', X_1, .., X_n]$ such that $Q(X_1', .., X_n')$
$- \lambda Q(X_1, .., X_n) = (X_1' - P_1)A_1 + \cdots + (X_n' - P_n)A_n$. Letting $X_i' = P_i$, we obtain $Q(P_1(X_1, ..., X_n), ..., P_n(X_1, ..., X_n)) = TQ(X_1, ..., X_n)$. [$(\Leftarrow)$] Conversely suppose

$Q(P_1(X_1, .., X_n), .., P_n(X_1, .., X_n)) = TQ(X_1, .., X_n)$. Then as $Q(X_1', .., X_n')$ is equal to $Q(P_1, .., P_n)$ modulo the ideal I, we get $Q(X_1', .., X_n') = \lambda Q(X_1, \dots, X_n)$ modulo I. This establishes the result. □

Proof of Theorem 5 Let Q be a polynomial in $\mathbb{R}[X_1, .., X_n]$. Such a polynomial Q is T-invariant if and only if $Q(P_1(X_1, .., X_n), ..,$
$P_n(X_1, .., X_n)) = T(X_1, .., X_n)Q(X_1, .., X_n)$, *i.e.*, if and only if $\mathscr{M}(Q) = \mathscr{L}(Q) \Leftrightarrow (\mathscr{M} - \mathscr{L})(Q) = 0_{\mathbb{R}[X_1,...,X_n]}$. Writing this in matrix equivalent terms we have $((M - L)Q = 0) \Leftrightarrow (Q \in Ker(M - L))$, and the result follows. □

Proof of Theorem 6 From linear algebra, we know that $M - L$ with a non-trivial kernel is equivalent to it having rank strictly less than the dimension $v(r)$ of $\mathbb{R}_r[x_1, \dots, x_n]$. This is equivalent to the fact that each $v(r) \times v(r)$ sub-determinant of $M_D - L_T$ is equal to zero. Those determinants are polynomials with variables $(t_1, .., t_{v(d-1)})$, which we will denote by $V_1(t_1, \dots, t_{v(d-1)}), \dots, V_s(t_1, \dots, t_{v(d-1)})$. From the form of L, this is zero when $(t_1, \dots, t_{v(d-1)}) = (0, \dots, 0)$. Hence, $M - L$ has its last column equal to zero, giving a common root for these polynomials, corresponding to the constant invariants. □

Proof of Theorem 7 Consider Theorem 5. The initiation step defines on $\mathbb{R}_r[x_1, \dots, x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1, \dots, u_n)$. Thus, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1, .., X_n]$ given by the kernel of I_u, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] \mid Q(u_1, \dots, u_n) = 0\}$. With initial conditions $(x_1(0) = u_1, \dots, x_n(0) = u_n)$, we are looking for a T-scale differential invariant in $\mathbb{R}_r[x_1, \dots, x_n]$ that belongs to the hyperplane $P(u_1, \dots, u_n) = 0$, *i.e.*, we are looking for Q in $Ker(M - L) \cap \{P \mid P(u_1, \dots, u_n) = 0\}$. □

Proof of Corollary 3 [$(\Rightarrow)$] If there is a T-scale invariant for any initial value, then the corresponding eigenspace has dimension at least 2. Indeed, if the space was of dimension only 1 (which is at least necessary to have T-invariants), taking any non-zero vector Q in the eigenspace (i.e. a T-invariant), Q should lie in any hyperplane of initial values, and so for every n-tuple $(u_1, \dots, u_n)$, one would have $Q(u_1, \dots, u_n) = 0$, hence $Q = 0$, which is absurd. [(⇐)] Any eigenspace of $M_D - L_T$ with dimension at least 2 will intersect any space given by any initial constraints. This establishes the result. □

Proof of Corollary 4 The right singular vectors corresponding to vanishing singular values of $\overline{M-L}$ span the null space of $\overline{M-L}$. The left singular vectors corresponding to the non-zero singular values of $\overline{M-L}$ span the range of $\overline{M-L}$. As a consequence, the rank of $\overline{M-L}$ equals the number of non-zero singular values which is the same as the number of non-zero elements in the matrix S. □

Proof of Theorem 8 [(⇒)] If $Q(X_1', .., X_n') - FQ(X_1, .., X_n)$ belongs to the fractional ideal J generated by the family $(X_1' - P_1/Q_1, \dots, X_n' - P_n/Q_n)$, then there exists a family $(A_1, \dots, A_n)$ of fractional functions in $\mathbb{R}(X_1', .., X_n', X_1, .., X_n)$ such that $Q(X_1', .., X_n') - FQ(X_1, .., X_n) = (X_1' - P_1/Q_1)A_1 + \cdots + (X_n' - P_n/Q_n)A_n$ Letting $X_i' = \frac{P_i}{Q_i}$ we obtain $Q(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}) = \lambda Q(X_1, \dots, X_n)$. [(⇐)] Conversely suppose $Q(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}) = FQ(X_1, .., X_n)$. Then as $Q(X_1', .., X_n')$ is equal to $Q(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n})$ modulo the ideal J, we get that $Q(X_1', .., X_n') = FQ(X_1, .., X_n)$ modulo J. And we have the result. □

Proof of Theorem 9 Let Q be a polynomial in $\mathbb{R}[X_1, .., X_n]$. In fact, a polynomial Q is T/Π^r-invariant if and only if $Q(P_1(X_1, .., X_n), .., P_n(X_1, .., X_n)) = T/\Pi^r(X_1, .., X_n)Q(X_1, .., X_n)$, which is equivalent to $\Pi^r Q(P_1(X_1, .., X_n),$
$.., P_n(X_1, .., X_n)) = T(X_1, .., X_n)Q(X_1, .., X_n)$, and this holds if and only if $(\mathscr{M}(Q) = \mathscr{L}(Q)) \Leftrightarrow ((\mathscr{M} - \mathscr{L})(Q) = 0_{\mathbb{R}[X_1,..,X_n]}$ Writing this in matrix equivalent terms we get $((M-L)Q = 0) \Leftrightarrow (Q \in Ker(M-L))$, and the result follows. □

Proof of Theorem 10 We first consider Theorem 9. The initiation step defines on $\mathbb{R}_r[x_1, \dots, x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1, \dots, u_n)$. Hence, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1, .., X_n]$ given by the kernel of I_u, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] \mid Q(u_1, \dots, u_n) = 0\}$. With initial conditions $(x_1(0) = u_1, \dots, x_n(0) = u_n)$, we are looking for a *strong*-scale differential invariant in $\mathbb{R}_r[x_1, \dots, x_n]$ that belongs to the hyperplane $P(u_1, \dots, u_n) = 0$, *i.e.*, we are looking for Q in $Ker(M-L) \cap \{P \mid P(u_1, \dots, u_n) = 0\}$. □

Proof of Corollary 5 [(⇒)] If there is a non-trivial F-scale invariant for any initial value, then the corresponding eigenspace has dimension at least 2. Indeed, if the space was of dimension only 1 (which is at least necessary to have F-invariants), taking any non-zero vector Q in the eigenspace (i.e. a F-invariant), Q should lie in any hyperplane of initial values, i.e. for every n-tuple $(u_1, \dots, u_n)$, one would have $Q(u_1, \dots, u_n) = 0$, hence $Q = 0$, which is absurd. [(⇐)] Any intersection between an eigenspace of M with dimension at least 2 will intersect any space given by any initial constraints. And we have the result. □

Proof of Theorem 11 Let $f_1^{(j)}, \dots, f_{n_j}^{(j)} \in K[X_1, \dots, X_n]$ be such that $I_j = (f_1^{(j)}, \dots, f_{n_j}^{(j)})$, for all j in $[1, k]$. Let $\beta \in (\otimes(I_1, \dots, I_k))$ Then there exists $e_1, \dots, e_{n_1 n_2 \dots n_k} \in K[X_1, .., X_n]$ such that $\beta = e_1\delta_1 + \cdots + e_{n_1 n_2 \dots n_k}\delta_{n_1 n_2 \dots n_k}$. Also, by construction of $\otimes(I_1, \dots, I_k)$ we know that for all $r \in [1, \dots, n_1 n_2 \dots n_k]$, $\delta_r \in \otimes(I_1, \dots, I_k)$. In other words, there is $(\alpha_1^{(r)}, \dots, \alpha_k^{(r)}) \in I_1 \times \cdots \times I_k$ such that $\delta_r = \Pi_{i=0}^k \alpha_i^{(r)}$. Then we have $\beta = \sum_{j=1}^{n_1 n_2 \dots n_k} [\lambda_j \, \Pi_{i=1}^k \alpha_i^{(j)}]$. Now, for all m in $[1, k]$, if I_m correspond to a precomputed inductive ideal of invariants associated to a transition τ_m at location l, then for all $j \in [1, n_1 n_2 \dots n_k]$ we have $\alpha_m^{(j)}(X1, \dots, Xn) = 0$. Hence, for all $j \in [1, n_1 n_2 \dots n_k]$ we get $\Pi_{i=1}^k \alpha_i^{(j)} = 0$. Finally, we obtain $\beta(X_1, \dots, X_n) = 0$ for all m in $[1, n_1 n_2 \dots n_k]$. In other words, $\beta(X_1, \dots, X_n) = 0$ is an algebraic assertion true at any step of the iteration of the loop for any transition τ_m that could possibly be taken. Then $(\beta(X_1, \dots, X_n) = 0)$ is an inductive invariant and we can conclude that $(\otimes(I_1, \dots, I_k))$ is an ideal of inductive invariants. □

References

[AV97] Arnaudiesa JM, Valibouze A (1997) Lagrange resolvents. J Pure Appl Algebra 117–118:23–40

[BBGL00] Bensalem S, Bozga M, Ghirvu J-C, Lakhnech L (2000) A transformation approach for generating non-linear invariants. Stat Anal Sympos 5:101–114

[BLS96] Bensalem S, Lakhnech Y, Saidi H (1996) Powerful techniques for the automatic generation of invariants. In: Alur R, Henzinger TA (eds) Proceedings of the 8th international conference on computer aided verification CAV, vol 1102, pp 323–335

[Buc96] Buchberger B (1996) Symbolic computation: computer algebra and logic. In: Frontiers of combining systems. Proceedings of the 1st international workshop, Munich, pp 193–220

[CC77] Cousot P, Cousot R (1977) Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Symposium of the principles of programming languages, pp 238–252. ACM Press, New York

[CC92] Cousot P, Cousot R (1992) Abstract interpretation and application to logic programs. J Logic Program 13(2–3):103–179
[Col75] Collins GE (1975) Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In: LNCS. Springer, New York
[Cou05] Cousot P (2005) Proving program invariance and termination by parametric abstraction, lagrangian relaxation and semidefinite programming. In: Conference on VMCAI, pp 1–24, Paris, LNCS, vol 3385, 17–19 January 2005
[CXYZ07] Chen Y, Xia B, Yang L, Zhan N (2007) Generating polynomial invariants with discoverer and QEPCAD. In: Formal methods and hybrid, real-time systems, pp 67–82
[Dij76] Dijkstra EW (1976) A discipline of programming. Prentince-Hall, London
[Fau99] Faugere J-C (1999) A new efficient algorithm for computing Grobner bases (f4). J Pure Appl Algebra 139(1–3):61–88
[Flo67] Floyd RW (1967) Assigning meanings to programs. In: Proceedings of the 19th symposium on applied mathematics, pp 19–37
[GT06] Gulwani S, Tiwari A (2006) Assertion checking over combined abstraction of linear arithmetic and uninterpreted functions. In: Sestoft P (ed) European symposium on programming, ESOP 2006, LNCS, vol 3924, pp 279–293
[GT08] Gulwani S, Tiwari A (2008) Constraint-based approach for analysis of hybrid systems. In: Proceedings of the 14th international conference on computer aided verification (CAV)
[Hoa69] Hoare CAR (1969) An axiomatic basis for computer programming. Commun ACM 12(10):576–580
[JKP06] Jebelean T, Kovacs L, Popov N (2006) Experimental program verification in the theorema system. Int J Softw Tools Technol Transf (STTT) (in press)
[Kap04] Kapur D (2004) Automatically generating loop invariants using quantifier elimination. In: Proceedings of the IMACS international conference on applications of computer algebra
[KJ06] Kovacs L, Jebelean T (2006) Finding polynomial invariants for imperative loops in the theorema system. In: Proceedings of the verify'06 workshop, pp 52–67, 15–16 August 2006
[Kov08] Kovacs L (2008) Reasoning algebraically about p-solvable loops. In: TACAS 2008, Proceedings of the 14th international conference on tools and algorithms for the construction and analysis of systems, LNCS, vol 4963, pp 249–264.
[Lan02] Lang S (2002) Algebra. Springer, New York
[MOS02] Müller-Olm M, Seidl H (2002) Polynomial constants are decidable. In: Static analysis symposium, LNCS, pp 4–19
[MP95] Manna Z, Pnueli A (1995) Temporal verification of reactive systems: safety. Springer, New York
[PC08] Platzer A, Clarke EM (2008) Clarke Computing differential invariants of hybrid systems as fixedpoints. In: Proceedings of the computer-aided verification, CAV 2008, Princeton, LNCS. Springer, New York
[PJ04] Prajna S, Jadbabaie A (2004) Safety verification of hybrid systems using barrier certificates
[RCK07a] Rodríguez-Carbonell E, Kapur D (2007) Automatic generation of polynomial invariants of bounded degree using abstract interpretation. Sci Comput Program 64(1):54–75
[RCK07b] Rodríguez-Carbonell E, Kapur D (2007) Generating all polynomial invariants in simple loops. J Symb Comput 42(4):443–476
[RM09] Rebiha R, Moura AV (2009) Automated malware invariant generation. In: 6th international conference on forensic computer science, ICoFSC2009 and ICCYBER2009 (best paper award)
[RM11a] Rebiha R, Moura AV (2011) Algebraic formal methods for invariant generation. In: Ph.D. Dissertation, Faculty of Informatics USI, University of Lugano, Switzerland, pp 1–209
[RM11b] Rebiha R, Moura AV (2011) Algebraic formal methods for invariant generation. In: Ph.D. Dissertation, Insitute of Computing UNICAMP, University of Campinas, Sao Paulo, pp 1–214
[RM11c] Rebiha R, Moura AV (2011) Semantic malware resistance using inductive invariants. Int J Forensic Comput Sci (IJoFCS0) (best paper award, 5)
[RMM08a] Rebiha R, Matringe N, Moura AV (2008) Endomorphism for non-trivial semi-algebraic loop invariant generation. In: Technical report TR-IC-08-31, Institute of Computing, University of Campinas
[RMM08b] Rebiha R, Matringe N, Moura AV (2008) Endomorphisms for non-trivial non-linear loop invariant generation. In: 5th international conference on theoretical aspects of computing, pp 425–439, LNCS
[RMM10] Rebiha R, Matringe N, Moura AV (2010) Generatin invariants for non-linear hybrid systems by linear algebraic methods. In: 17th international static analysis symposium, SAS2010, LNCS
[Sch86] Schrijver A (1986) Theory of linear and integer programming. Wiley, New York
[SJ05] Stein W, Joyner D (2005) SAGE: system for algebra and geometry experimentation. ACM SIGSAM Bull 39(2):61–64
[SSM04a] Sankaranarayanan S, Sipma H, Manna Z (2004) Constructing invariants for hybrid system. In: Hybrid systems: computation and control (HSCC), LNCS, vol. 2993, pp 539–554. Springer, New York
[SSM04b] Sankaranarayanan S, Sipma HB, Manna Z (2004) Non-linear loop invariant generation using grobner bases. In: POPL '04, Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on principles of programming languages, pp 318–329. ACM Press, New York
[Tiw08] Tiwari A (2008) Generating box invariants. In: Proceedings of the 11th international conference on hybrid systems: computation and control (HSCC)
[Wei97] Weispfenning V (1997) Quantifier elimination for real algebra—the quadratic case and beyond. Appl Algebra Eng Commun Comput 8(2):85–101

Received 7 November 2012
Revised 4 January 2015
Accepted 2 February 2015 by E. Allen Emerson
Published online 18 March 2015