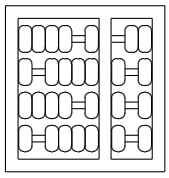


INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Generating Invariants for Non-linear Loops by Linear Algebraic Methods

Rachid Rebiha *Arnaldo V. Moura*
Nadir Matringe

Technical Report - IC-13-04 - Relatório Técnico

February - 2013 - Fevereiro

Generating Invariants for Non-linear Loops by Linear Algebraic Methods

Rachid Rebiha* Arnaldo Vieira Moura† Nadir Matringe ‡

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 Grobner basis computation, quantifier elimination or cylindrical algebraic decomposition. 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 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 loop invariant ideals. As far as it is our knowledge, it is the first invariant generation methods that handle multivariate fractional loops. Our algorithm also incorporates a strategy to guess the degree bounds which allow for the generation of ideals of non-trivial invariants.

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 that addresses various deficiencies of other state-of-the-art non-linear invariant generation methods. More generally, we provide mathematical techniques and design efficient algorithms to automate the discovery and the strengthening of non-linear interrelationships among the variables of programs containing non-linear loops, and which give rise to multivariate polynomial and fractional relationships.

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,

*Instituto de Computação, Universidade Estadual de Campinas, 13081970 Campinas, SP. Pesquisa desenvolvida com suporte financeiro da FAPESP, processo 2011089471

†Instituto de Computação, Universidade Estadual de Campinas, 13081970 Campinas, SP.

‡Université de Poitiers, Laboratoire Mathématiques et Applications and Institue de Mathematiques de Jussieu Université Paris 7-Denis Diderot, France.

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 assertion directly to prove program properties, or to provide supporting lemmas that can be used to established 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 [Flo67, Hoa69] inductive assertion technique, 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 every 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 high number of Gröbner Bases computation [SSM04b], first-order quantifier elimination [Wei97, Col75], or cylindrical algebraic decomposition [CXYZ07]. Invariants can also be computed as fixed points of operations on ideals by fixed point techniques [RCK07a] and using abstract interpretations [CC92, CC77] framework 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 invariant generation in the presence of non linear loops. Other methods [KJ06, Kov08] attempt to generate invariants from a restricted class of P-solvable loops. Their methods use techniques from algebra and combinatorics, like Gröbner bases [JKP06], variable elimination, algebraic dependencies and symbolic summation, and so also incurr in high computational complexities.

More recent approaches have been constraint-based [SSM04b, RCK07a, Kap04, RCK07b, SSM04a, SA08, 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, SYH96, CXYZ07, Kov08, KJ06, Cou05, MOS02, RCK07b, SA08, A. 08, 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 a more general form 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 of specific endomorphisms and initial constraints.

- We provide general sufficient conditions guaranteeing the existence and allowing the computation of invariant ideals. Further, our approaches do 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.

- Our techniques comprise 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 non-linear systems of equations, rendering unfeasible their resolution; see Section 4.3. We, therefore, propose a computational method of much lower time complexity than other present approaches based on fixed point computation, or on constraint-based approaches.

- Also, 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 Grobner basis computations or quantifier eliminations, for example.

Example 1.1. (Motivational Example) *Consider the following program loop:*

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

The most recent and best performing techniques for program verifications and static analysis are not able to produce any conclusion that could be somehow related to the values of the variables x and y because the semantic of the two instructions inside the loop relies non-linear arithmetic. Such non-linearities are presently recognized by industry and academia as a critical bottleneck for automatic program verification and static analysis. In this article, we introduce new symbolic techniques with fast numerical approaches that can be used in these situations. Our methods can directly compute $\{x^2, x*y-x, y^2-2y+1\}$ as a basis for the vector space of invariants, and we note that all elements in this space would 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, $y_0(1-y_0)x^2 + xy - x + y^2 - 2y + 1 = 0$. Such invariants are beyond the reach of other current invariant generation techniques. □

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

2 Ideals of Polynomials and Inductive Assertions

We will use the following notations. The ring of multivariate polynomials over the set of variables $\{X_1, .., X_n\}$ will be indicated by $\mathbb{K}[X_1, .., X_n]$. We will denote by $\mathbb{R}_d[X_1, .., X_n]$ the ring of multivariate polynomials of degree at most d over the set of real variables $\{X_1, .., X_n\}$. We will write $Vect(v_1, ..., v_n)$ for the vector space generated by the basis $v_1, ..., v_n$. We will write $Ker(M)$ and $Rank(M)$ for the kernel and rank, respectively, of a

(vector space) morphism M. A primed x' will refer to the next state value of a variable x, after a transition is taken.

2.1 Ideals of Polynomials

Definition 2.1. *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

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

Definition 2.2. *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*.

2.2 Inductive Assertions and Invariants

We use *transition systems* as representation of imperative programs and *automata* as their computational models.

The contribution and novelty in our approach clearly set it apart from those in [SSM04b] as their constraint-based techniques are based on several Grobner Basis computations and on 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.

Definition 2.3. *A* transition system *is given by $\langle V, L, \mathcal{T}, l_0, \Theta \rangle$, where*

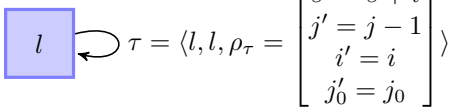
- *V is a set of variables,*
- *L is a set of locations and $l_0 \in L$ is the initial location.*
- *A* state *is given by an interpretation of the variables in V.*
- *A* transition *$\tau \in \mathcal{T}$ is given by a tuple $\langle l_{pre}, l_{post}, \rho_\tau \rangle$, where l_{pre} and l_{post} 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 said to be affine *when ρ_τ is an affine form. And it is said to be* algebraic *when ρ_τ is an algebraic form.*

□

Example 2.1. *Consider the program depicted at the left below, for multiplying two numbers. Its computational model is described by the automaton at the right:*

```
int s, i, j, j_0;
//initialization
(s=0)&&(j=j_0)
       ...
 While (...){
   s := s+i;
   j := j-1;
  }
```



with $V = \{s, i, j, j_0\}$, $\Theta = (s = 0 \wedge j = j_0)$, $l_0 = l$, $L = \{l\}$ and $\mathcal{T} = \{\tau\}$.

□

Definition 2.4. *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 wnat to say that every possible transition preserves the invariant, when specific consecution conditions hold. That is, in order to generate loop invariants one needs to discover *inductive* assertions.

Definition 2.5. *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 3.1. *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 des 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 $\mathbb{R}[X_1, ..., X_n]$. Also, the T polynomials can be understood as *template multiplicative factors*. In other words, 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 fractionals $\frac{T}{Q}$ 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 use 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 ρ_τ:

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

where $L_i(X_1, ..., X_n) = \sum_{k=1}^{n} c_{i,k-1}X_k + c_{i,k}$ are affine or linear forms.

4.1 Generating λ-scale invariants

We have the following λ-scale invariant characterization.

Theorem 4.1. *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*

$$Q(L_1(X_1,..,X_n),..,L_n(X_1,..,X_n)) = \lambda Q(X_1,..,X_n). \tag{2}$$

□

In this case, $Q \in \mathbb{R}[X_1,..,X_n]$ is of degree r. We show that for good choices of λ there always exists such a λ-invariant that is also not 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 Eq. (2) into linear algebra, consider the morphism

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

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

Example 4.1. (Running example) *Consider the following loop $\tau = \langle l_i, l_i, \rho_\tau \rangle$ with*

$$\rho_\tau = \begin{bmatrix} x_1' = 2x_1 + x_2 + 1 \\ x_2' = 3x_2 + 4 \end{bmatrix}. \tag{3}$$

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 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 the basis B_1, expressing the results in the same basis. For the first column of M we first 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*

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

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

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 4.1, we have

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

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

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

where $\mathscr{I}$ is the identity endomorphism and I is the associated identity matrix of $\mathbb{R}_r[X_1, ..X_n]$. 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 4.2. *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. $\square$

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 gives 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 4.2. *Looking at the eigenvalues of the matrix* M *of the previous running Example 4.1, if we fix* λ *to be* 4*, we get that the corresponding eigenspace is generated by the vector* $(1, -2, 1, -6, 6, 9)^\top$*. As a* λ*-invariant polynomial* Q *for constant-scale consecution with parameter* 4*, we get the following output from our prototype* Ideal_Inv_Gen *(see also Section 9):*

```
Constant scaling discrete step
Lambda = 4 Eigenspace
{{1, -2, 1, -6, 6, 9}}
```

Interpreted in the canonical basis of $\mathbb{R}[x_1, x_2]$*, the associated* 4*-invariant is*

$$Q(x_1, x_2) = 1{x_1}^2 - 2x_1x_2 + {x_2}^2 - 6x_1 + 6x_2 + 9.$$

□

Example 4.3. *(General Case for 2 Variables) 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 charateristic 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 monomial* x_1^2*,* x_1x_2 *and* x_2r *must be non null for such an eigenvector.* □

Corollary 4.1. *Let* M *be the matrix introduce 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 $u_1, ..., u_n$ be the initial values of $X_1, ..., X_n$. For the initial step we need $Q(u_1, ..., u_n) = 0$. We have $P \mapsto P(u_1, ..., u_n)$ as a linear form in $\mathbb{R}_r[X_1, ..., X_n]$. Hence initial values correspond to a hyperplane in $\mathbb{R}_r[X_1, ..., X_n]$, given by the kernel of $P \mapsto P(u_1, ..., u_n)$. Now, if we add the initiation step, $Q(X_1, ..., X_n) = 0$ will be an inductive invariant (see Definition 2.4) 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 4.3. *A polynomial Q in $\mathbb{R}_r[X_1, .., X_n]$ is an inductive invariant for the affine loop (see Definition 2.5) 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 4.2. *There will be a non-null invariant polynomial for any given initial values if and only if there exists an eigenspace of M with dimension at least 2.* □

Example 4.4. *We return to running Example 4.1. 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 basis $(1, -2, 1, -6, 6, 9)^\top$, E_6 has basis $(0, 1, -1, 2, -5, 6)^\top$, E_9 has basis $(0, 0, 1, 0, 4, 4)^\top$, E_2 has basis $(0, 0, 0, 1, -1, -3)^\top$, E_3 has basis $(0, 0, 0, 0, 1, 2)^\top$, and E_1 has 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 conditons and consider general conditions in the sequel. Theorem 4.3 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 4.5. *We study the following transition system [SSM04b], corresponding to the multiplication of 2 numbers, and where the transition considered 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:*

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

- *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 4.2 applies. For example, the vector*

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

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, Sipma and Manna 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

Let ρ_τ be the algebraic transition relation

$$\rho_\tau \equiv \begin{bmatrix} x'_1 = P_1(x_1, \ldots, x_n) \\ \vdots \\ x'_m = P_m(x_1, \ldots, x_n) \end{bmatrix}, \tag{4}$$

where each polynomial P_i has a degree greater than 1.

Example 4.6. *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

$\{x' - x(y+1), y' - y^2\}$, *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 discrete systems.

5.1 T-scale invariant generation

Consider an algebraic transition system:

$$\rho_\tau \equiv \begin{bmatrix} X_1' = P_1(X_1, .., X_n) \\ \vdots \\ X_n' = P_n(X_1, .., X_n) \end{bmatrix}, \tag{5}$$

where the P_i's are in $\mathbb{R}[X_1, .., X_n]$. We have the following T-scale discrete invariant characterization.

Theorem 5.1. *A polynomial* Q *in* $\mathbb{R}[X_1, .., X_n]$ *is a* T*-scale discrete invariant for polynomial-scale consecution with a* parametric polynomial $T \in \mathbb{R}[X_1, ..., X_n]$ *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 5.1. *Reconsider Example 4.6. 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*:*

$$\begin{array}{lll} a_0 = b_0a_0 & 0 = b_2a_5 + b_3a_4 & a_3 = b_1a_4 + b_2a_3 + b_3a_1 \\ a_1 = b_0a_1 & 0 = b_0a_4 + b_2a_2 & a_4 = b_0a_5 + b_2a_4 + b_3a_2 \\ a_2 = b_0a_2 & a_3 = b_1a_5 + b_3a_3 & a_1 = a_3b_0 + b_1a_2 + b_2a_1 \\ a_5 = b_3a_5 & a_0 = b_1a_3 + b_3a_0 & \\ 0 = b_1a_0 & 2a_0 = b_1a_1 + b_2a_0. & \end{array}$$

Now we describe a decision procedure for parameter valuation. 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* $b_1a_0 = 0$. *For the same reason, choose* $b_1 = 0$. *Then we have* $b_2a_0 = 2a_0$. *As a direct consequence,* b_2 *is set to* 2. *Since equation* $b_3a_0 = a_0$ *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 the generating of* T*-invariants. Now, we add 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* $(y_0(1 - y_0), 1, 1, -1, -2, 1)^\top$. *Here,* $y_0(1-y_0)x^2+xy+y^2-x-2y+1 = 0$ *is an invariant. Note that* $T(x, y) = y^2+y+1$. □

Remark 5.1. *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 with 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 $\lambda_0, ..., \lambda_s$, 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

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

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

$$\mathscr{L} : \left\{ \begin{array}{rcl} \mathbb{R}_r[X_1, \ldots, X_n] & \to & \mathbb{R}_{dr}[X_1, \ldots, X_n] \\ P & \mapsto & TP. \end{array} \right.$$

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,

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

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 5.2. *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.* □

Again, the last column of M is $(0, ..., 0, 1)^\top$. 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 V_r. This is equivalent to the fact that each $v(r) \times v(r)$ subdeterminant of $M - L$ is equal to zero [Lan02]. Those determinants are polynomials with variables $(\lambda_0, \lambda_1, \cdots, \lambda_s)$, which we will denote by $V_1(\lambda_0, \lambda_1, \cdots, \lambda_s), \ldots, V_s(\lambda_0, \lambda_1, \cdots, \lambda_s)$.

Theorem 5.3. *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 5.2. *This theorem provides us with important existence results. But there is a more practical way of computing invariant ideals without computing common roots and subdeterminants. We will examine that in the next section.* □

Example 5.2. *(Loop with two variables, T-scale invariant of degree 2) 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 subdeterminant obtained by canceling 9 lines of $M - L$. They will be polynomials of degree less than 6 in the variables $(\lambda_0, ..., \lambda_5)$. Then, L is such that $M - L$ will be of degree less than 6 if and only if $(\lambda_0, ..., \lambda_5)$ are roots of each of those polynomials. □

Remark 5.3. *In many cases, it is easy to find a matrix L such that $M - L$ has a non trivial kernel. We describe two decidable classes: (i) suppose that in the previous case, c_2, c_4 and c_5 are null, then one can choose $(\lambda_0, \ldots, \lambda_s)$ in order to make the first column zero; and (ii) the third column can be canceled using good choices for the λ_i's, if d_0, d_3 and d_5 are zero.* □

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 $P \mapsto P(u_1, ..., u_n)$. Hence, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1, .., X_n]$, given by the kernel of $P \mapsto P(u_1, ..., u_n)$, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] \mid Q(u_1, \ldots, u_n) = 0\}$.

Theorem 5.4. *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 $L \neq 0$ (the 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 $Ker(M - L)$ would intersect any initial (semi-)hyperplane. We can state the following Corollary, important in practice.

Corollary 5.1. *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 the template multiplicative in T, such that $Ker(M - L)$ has dimension at least 2.

5.4 Example

Example 5.3. *(Running example) Consider the following transition:*

$$\tau = \langle l_i, l_j, \rho_\tau \equiv \begin{bmatrix} x' = xy + x \\ y' = y^2 \end{bmatrix} \rangle.$$

- *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 $\lambda_0, ..., \lambda_5$ 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 also 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*

$$M - L = \begin{pmatrix} -\lambda_0 & 0 & 0 & 0 & 0 & 0 \\ -\lambda_1 & -\lambda_0 & 0 & 0 & 0 & 0 \\ 1-\lambda_2 & -\lambda_1 & -\lambda_0 & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & -\lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda_2 & 0 & 0 & 0 \\ -\lambda_3 & 0 & 0 & -\lambda_0 & 0 & 0 \\ 2-\lambda_4 & -\lambda_3 & 0 & -\lambda_1 & -\lambda_0 & 0 \\ 0 & 1-\lambda_4 & -\lambda_3 & -\lambda_2 & -\lambda_1 & 0 \\ 0 & 0 & -\lambda_4 & 0 & -\lambda_2 & 0 \\ 1-\lambda_5 & 0 & 0 & -\lambda_3 & 0 & -\lambda_0 \\ 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_3 & -\lambda_1 \\ 0 & 0 & -\lambda_5 & 0 & 1-\lambda_4 & -\lambda_2 \\ 0 & 0 & 0 & 1-\lambda_5 & 0 & -\lambda_3 \\ 0 & 0 & 0 & 0 & -\lambda_5 & -\lambda_4 \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_5 \end{pmatrix}.$$

- *Step 2: We then 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.*

- *Step 3: Now matrix $M - L$ satisfies the hypotheses of Theorem 5.2. So, there will always be invariants, whatever the initial values. We compute the basis of $Ker(M-L)$:*

```
Polynomial scaling discrete step
T(x,y) = y^2 + 2 y + 1
Module of degree 6 and rank 3 and Kernel of dimension 3
{{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 $\mathbb{R}_2[x, y]$:

```
Basis of invariant Ideal
{x^2, x y - x, y^2 - 2 y + 1}
```

We have thus 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*

$$y_0(1 - y_0)x^2 + xy - x + y^2 - 2y + 1 = 0.$$

□

6 Obtaining optimal degree bounds for discrete transition systems

In order to guarantee the existence of non-trivial invariants, we are looking 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 the computation of invariants.

As input we have r, the candidate degree for the set of basis invariant elements, and $P_1, ..P_n$, the n polynomials given by the transition relation in considered loop. We first compute d, the maximal degree of the P_i's as can be seen by `Max_degree(`$\{P_1, ..., P_n\}$`)`, at line 4. Then, we detail the cases were the transitions are defined by non-linear systems, i.e. $d \leq 2$. Then, we define T as a polynomial of degree $dr - r$ in its canonical form,*i.e.* with parameterized coefficients. See `Template_Canonical_Form(`$dr-r$`)`, at line 7. We can, then, build a decision procedure to assign values to the coefficients of T in such a way that $Ker(M - L) \neq 0$. 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 terms in M to parameters in L. See, at line 10, the call to `Reduce_Rank_Assigning_Values(`$M-L$`)`. By so doing we can zero or identify some columns or lines of $M - L$. Next, we determine whether the matrix obtained, $\overline{M - L}$, has a trivial kernel by first computing its rank and then checking if $\mathbf{Rank(}\overline{M - L}\mathbf{)} < \mathbf{Dim(}R_r[X_1, .., X_n]\mathbf{)}$ holds, at line 11. When $\overline{M - L}$ has a trivial kernel, we can increase the degree r of the invariants until Theorem 5.2 (or Corollary 5.1) 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(`$r + 1, P_1, ..., P_n, X_1, ..., X_n$. 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, we compute and output the basis

Algorithm 1: Ideal_Loop_Inv_Gen$(r, P_1, ..., P_n, X_1, ..., X_n)$

/*Guessing the degree bounds for discrete transisions.*/;
Data: r is the candidate degree for the set of basis invariants elements we are looking for, $P_1, ..P_n$ the n are polynomials given by the considered loop, and $X_1, ..X_n \in V$
Result: $Ideal\_Inv$, a basis of ideal of invariants.
begin

```
1   int d;
2   Template T;
3   Matrix M, L;
4   d ⟵ Max_degree({P_1, ..., P_n});
5   /*d is the maximal degree of P_i's*/;
6   if d >= 2 then
7       T ⟵ Template_Canonical_Form(dr − r);
8       M ⟵ Matrix_D(r, dr, P_1, ..., P_n);
9       L ⟵ Matrix_L(r, dr, T);
10      M̄ − L ⟵ Reduce_Rank_Assigning_Values(M − L);
11      if Rank(M̄) >= Dim(R_r[X_1, .., X_n]) then
12          return Ideal_Loop_Inv_Gen(r + 1, P_1, ..., P_n, X_1, ..., X_n);
13          /*We need to increase the degree r of candidates invariants.*/;
14      else
15          return Nullspace_Basis(M̄ − L);
16          /*There exists an ideal of invariants that we can compute*/;
17  else
18      ... /*We refer to our previous work for constant scaling.*/;
```

of the nullspace of the matrix $\overline{M - L}$, in order to construct an ideal basis for non-trivial invariants. See `Nullspace_Basis`, at line 15. For the latter, we use well-known state-of-the-art algorithms, for example those that Mathematica 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 $\overline{M}$ by a two step procedure. 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 implementation of the algorithm we could rewrite Corollary 5.1 as follows.

Corollary 6.1. *Let $\overline{M-L} = USV$ be the singular value decomposition of matrix $\overline{M-L}$ 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.* □

Remark 6.1. *It is important to emphasize that eigenvectors 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:

$$\begin{bmatrix} X_1' = \frac{P_1(X_1,..,X_n)}{Q_1(X_1,..,X_n)} \\ \vdots \\ X_n' = \frac{P_n(X_1,..,X_n)}{Q_n(X_1,..,X_n)} \end{bmatrix}, \tag{6}$$

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 7.1. *(F-scale invariant characterization) 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.*, $i_1+..+i_n \leq r$. Then,

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

But as $Q_j^{i_j}$ divides Π^{i_j}, for all j, we see that $Q_1^{i_1}...Q_n^{i_n}$ divides $\Pi^{i_1+\ldots+i_r}$, which divides Π^r. We deduce that $\Pi^r Q(P_1/Q_1,\ldots,P_n/Q_n)$ 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

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

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 will divides Π^r, and we can suppose that it has denominator Π^r. So, let F be of the form T/Π^r, and we just argued that this constraint is weak.

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

$$\mathscr{M} : \left\{ \begin{array}{rcl} \mathbb{R}_r[X_1, \ldots, X_n] & \rightarrow & \mathbb{R}_{nrd}[X_1, \ldots, X_n] \\ Q & \mapsto & \Pi^r Q(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}). \end{array} \right.$$

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

$$\mathscr{L} : \left\{ \begin{array}{rcl} \mathbb{R}_r[X_1, \ldots, X_n] & \rightarrow & \mathbb{R}_{nrd}[X_1, \ldots, X_n] \\ Q & \mapsto & TQ. \end{array} \right.$$

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

Theorem 7.2. *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 5.3 and 5.4. For the initiation step, we have a hyperplane in V_r. 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 so 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 7.3. *(Non trivial invariants using fractional scale consecution) 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 corresponding to vectors in the intersection.* □

We also have the following important corollary.

Corollary 7.1. *(Non trivial invariants using fractional-scale consecution and for any initial value) We will have a non-trivial invariant for any non-trivial initial value if there exists a matrix L such that the dimension of $Ker(M - L)$ is at least 2.* □

Example 7.1. *(Running example) Consider the system*

$$\rho_\tau \equiv \begin{bmatrix} x_1' = \frac{x_2}{(x_1+x_2)} \\ x_2' = \frac{x_1}{(x_1+2x_2)} \end{bmatrix}. \tag{7}$$

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:

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

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

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

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*

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

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 cancles out and the kernel will be equal to* $Vect(0, 1, 0)$*. Now, Corollary 7.1 applies to* $M - L$*:*

```
Fractional scaling discrete step
T(x,y) / Q(x,y) = 1 / ((x + y) (x + 2 y))^2
Module of degree 3 and rank 1 and Kernel of dimension 2
{{0, 1, 0}}
Basis of invariant Ideal
{ x y }
```

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.* □

8 Branching conditions and nested loops

We have generated a basis of a vector space which describes 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. This approach is a sound, but not complete, way of computing ideals for global invariants, and it has a low computational complexity. In order to take into account initial conditions we intersect these vector spaces of invariants with the initial semi-hyperplanes

deduced from the isomorphism associated with initial requirements. Here, 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: $\rho_{\tau_1} \equiv [x'_1 = F_{1,[I_1;I_2;I_4;]_\circ}(x_1, ...x_n), \ldots, x'_n = F_{n,[I_1;I_2;I_4;]_\circ}(x_1, ..., x_n)]$ and $\rho_{\tau_2} \equiv [x'_1 = F_{1,[I_1;I_3;I_4;]_\circ}(x_1, ...x_n) \ldots, x'_n = F_{n,[I_1;I_3;I_4;]_\circ}(x_1, ..., x_n)]$, with $[;;]_\circ$ denoting our operator based on separation rewriting rules and used to compose blocks of instructions. We first independently generate the ideals of invariants $\xi_1 = (\mu_1, ..., \mu_n)$ and $\xi_2 = (\kappa_1, ..., \kappa_p)$ for the respective transitions τ_1 and τ_2. Any element $\mu_i \in \xi_1$ refers to an inductive invariant $\mu_i(X_1, ..., X_n) = 0$ corresponding to the *partial loops* described by transition τ_1. Similarly, any $\kappa_i \in \xi_2$ refers to an inductive invariant $\kappa_i(X_1, ..., X_n) = 0$ for the loop described by transition τ_2. Then we can take $\mu_i(X_1, ..., X_n) * \kappa_i(X_1, ..., X_n) = 0$ as global loop invariants, since these invariant will remain true in any sequence of transitions during the execution of the loop.

Theorem 8.1. *Let $I = \{I_1, ..., I_k\}$ a set of ideals in $\mathbb{R}[X_1, ..., X_n]$ such that $I_j = ({f^{(j)}}_1, ..., f^{(j)}_{n_j})$ where $j \in [1, k]$. Let $\otimes(I_1, ..., I_k) = \{\delta_1, ..., \delta_{n_1 n_2 ... n_k}\}$ be such that all elements δ_i in $\otimes(I_1, ..., I_k)$ 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. Now, if all l_j describe the same location or program point l, then we have several transitions* looping *at the same point. Thus we can obtain an encoding of possible execution paths of a loop containing* conditional statements*. It is clear then that $\otimes(I_1, ..., I_k)$ is an ideal of non-trivial non-linear invariants for the entire loop located at l.* □

We deal with loop conditions using the same methods that we propose to handle initiation conditions. We know, for instance, that if our Corollary 5.1 holds, then there exist invariants for any (semi-)hyperplane that could be induced by the loop conditions. We illustrate this point in Figure 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 8.1, 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.

Example 8.1. *Consider the following loop.*

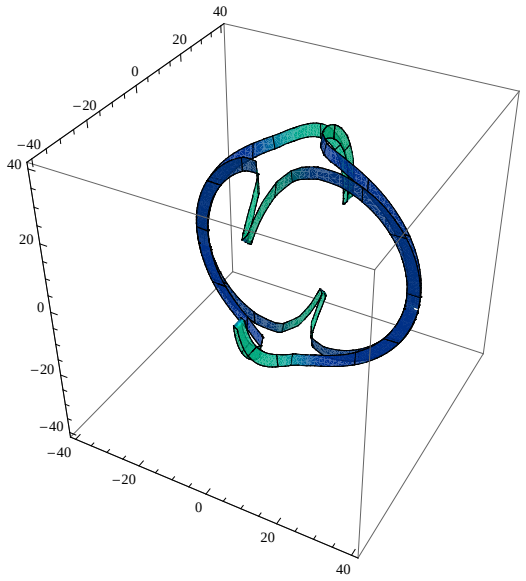


Figure 1: Intersection between the conditional loop: $800 < (x-5)^2 + (y-5)^2 + (y_0-5)^2 < 1000$ and the invariant $y_0(1-y_0)x^2 + xy - x + y^2 - 2y + 1 = 0$ from the invariant ideal $(\{x^2, xy - x, y^2 - 2y + 1\})$ computed for the running example 5.3.

```
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 and obtain:*

```
If_1 :
Fractional scaling discrete step
T(x,y) / Q(x,y) = 1 / ((x + y) (x + 2 y))^2
...
Basis of invariant Ideal
{ x y }
...
```

See Example 7.1 for more details. Then, we compute the invariant and get

```
Else_1 :
...
Basis of invariant Ideal
{ u_0z^2-u_0^2z^2+zu+u^2-z-2u+1, ... }
...
```

corresponding to the other alternative transition τ_2 of the loop, namely, the `Else`*. The* `Join` *operator now retunrs the invariants:*

```
While_1 :
...
{ xyu_0z^2-u_0^2z^2+xyzu+xyu^2-xyz-2xyu+xy, ... }
...
```

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 steps described remain polynomial. □

Example 8.1 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 Discussion and some experimental results

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, SYH96, CXYZ07, Kov08, KJ06, Cou05, MOS02, RCK07b, SA08, PC08]. In order to better understand the difficulties they raise, we first need some details on Gröbner basis and the ideal membership problem.

Consider a multivariate polynomial, $Q = \sum_{i_1,..,i_n} a_{i_1,..,i_n} {x_1}^{i_1} ... {x_n}^{i_n}$, where the coefficients $a_{i_1,...,i_n}$ are in a field K. How do we know if it is in an ideal I of $K[X_1, ..., X_n]$? This is known as the *Ideal membership problem*. To handle it we can use a Gröbner basis $G = \{g_1, ..., g_s\}$ for I. There are algorithms that compute such bases as long as we know a finite generating bases of 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 $NF_G(Q) = \sum_{i_1,..,i_n} f(a)_{i_1,..,i_n} {x_1}^{i_1} ... {x_n}^{i_n}$, where $f(a)_{i_1,..,i_n}$ is a combinations of the coefficients $a_{i_1,..,i_n}$. Then the statement $(Q \in I)$ is equivalent to the assertion $(NF_G(Q) = 0)$, that is, all the coefficients $f(a)_{i_1,...,i_n}$ 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 of these techniques, is to obtain the reduction $NF_G(Q)$. An important obstacle faced here is that all known algorithms for computing Gröbner basis and for the 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 the system of equations $(NF_G(Q) = 0)$, and attempt to solve it directly. Moreover, we have shown (see Section 4.3) that as soon as the loop contains a non-linear instruction, the constraints considered in their final step is a non-linear equation systems.

In terms of performance and efficiency, we succeeded in reducing the non-linear loop invariant generation problem to a linear algebraic problem, i.e. the computation of eigenspaces of specific morphisms. Our techniques have few computational steps: we compute first some specific matrices and we then compute their nullspaces. Each step performed by our techniques remains of polynomial complexity. Further, our approaches do not generate an invariant at a time. Instead we generate an ideal of invariants which is an enormous (infinite) structure. We also handle fractional systems and our algorithm incorporates a strategy to guess 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 invariant ideals. Note that these conditions could be directly used by any invariant generation method.

We clearly see that each main step of these approaches incurs in a doubly exponential number of elementary computations. 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. Moreover, as we have shown (see Section 4.3) that 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.

We have coded a prototype system, called Ideal_Inv_Gen, that implements the algorithms described by our techniques. The third column in Table 1(a) 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 1(a) gives some existential results which, we note, can also be used by other constraint-based approach or reachability analysis methods.

We have also used it to obtain some experimental results that attest to the effectiveness and scope of our methods. In Table 1(b) we list some of these experimental results. We can see that our methods efficiently handle a large number of non-linear examples treated elsewhere in the literature. The experiments 5, 6 and 7 are from [SSM04b] and are basically linear systems. Moreover, the constraint-based invariant generation approach would be construct only a single linear inductive invariant. Section 4.3 gives more details on the constraint-based standard approaches limits. In contrast, by using our methods, we were able to generate vectorial spaces of non-trivial non-linear invariants in a polynomial number of computational steps, using mostly a constant scaling approximation. Experiments 1—4 and 5—19, listed in Table 1(b), involve non-linear systems most of which can be shown to be beyond the limits of other recent approaches. Those results show the strength of our approach for generating non-linear invariants for non-linear systems.

As a more applied motivation, our techniques can be made to bear on new domains

Table 1: Examples and Experimental Results

(a) Linear algebraic problems and consecution approximations

Prog. Loop	Aprox.Consec.	Lin. Algeb. Prob.	Existence Cond.
Affine/lin. inst.	Lambda Scaling	Eigenspaces	$Ker(M_D) \geq 2$ for any init. cond., and $Ker(M_D) \neq \emptyset$ 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	$Ker(M_\Pi - L_T) \geq 2$ for any init. cond., and $Ker(M_\Pi - L_T) \neq \emptyset$ otherwise.

(b) Experimental results: Basis of invariant ideals obtained automatically by our prototype Ideal_Inv_Gen

Non-Linear Loop Prog.			Scaling	CPU/s
1 - From [RMM08b] Example 2.			*Lambda*	0.04
2 - From [RMM08b] Example 3.			*Lambda*	0.08
3 - From [RMM08a] Example 1.			*Lambda*	0.04
4 - From [RMM10] Example 5.			*Polynomial.*	0.70
5 - From [SSM04b] Example 1, Section (5).			*Lambda*	0.05
6 - From [SSM04b] Example 2, Section (5).			*Lambda*	0.18
7 - From [SSM04b] Example 3, Section (5).			*Polynomial*	0.31
8 - From [RMM10] Example 6.			*Fractional*	0.90
9 - From [RMM08b] Example 6.			*Fractional*	0.90
10 - From [RMM08b] Example 4.			*Polynomial*	0.70
11 - From [RMM08b] Example 5.			*Polynomial*	1.65
12 - From [RMM08a] Example 2.			*Lambda*	0.04
13 - From [RMM08a] Example 5.			*Polynomial*	1.84
14 - From [RM11a] Example 2.			*Lambda*	0.04
15 - From [RM11b] Example 4.			*Lambda*	0.03
16 - From [RM11a] Example 6.			*Lambda*	0.08
17 - From [RM11b] Example 9.			*Polynomial*	1.23
18 - From [RM11a] Example 10.			*Polynomial*	1.17
19 - From [RM11b] Example 11.			*Fractional*	1.21

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 system. Invariants generated over malware codes are strong semantic aware signatures that can be used to analyse and identify intrusions caused by malicious such code. These new approaches could form intrusion detection systems where an alarm is a proof of abnormal behavior caused by the violation of a pre-computed invariant induced by the intrusion. These binary codes give rise to non-linear arithmetic and the methods described here allow, as we have shown, for the generations of complex and precise invariants. And the more the complex the invariant is, the harder it will be to morph the corresponding signatures in an automatic way.

10 Conclusions

Our primary goal and motivation were to provide invariant generation methods for static analysis and 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, more precisely, to the computation of eigenspaces of these morphisms. We also treated fractional systems and our algorithms incorporate a strategy to guess degree bounds for candidate invariants, thus allowing for the automatic generation of non-trivial invariants.

These techniques gave rise to algorithms of much lower time complexity than other modern approaches that incur in computations which are of a doubly exponential time complexity. By contrast, our techniques induce algorithms of polynomial time complexity.

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

We also noted that our techniques could be combined with other formal verification methods and their associated tools. As a case in point, we composed our techniques with formal methods that treat logics with uninterpreted functions [GT06], such as logics that handle function calls and operating system calls.

References

[A. 08] A. Tiwari. Generating box invariants. In *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control HSCC*, 2008.

[BBGL00] S Bensalem, M Bozga, J-C Ghirvu, and L Lakhnech. A transformation approach for generating non-linear invariants. *Static Analysis Symposium*, 5:101–114, June 2000.

[Buc96] Bruno Buchberger. Symbolic computation: Computer algebra and logic. In *Frontiers of Combining Systems: Proceedings of the 1st Int. Workshop, Munich (Germany)*, pages 193–220, 1996.

[CC77] P Cousot and R Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conf. Record of the 4th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977. ACM Press, NY.

[CC92] P. Cousot and R. Cousot. Abstract interpretation and application to logic programs. *Journal of Logic Programming*, 13(2–3):103–179, 1992.

[Col75] G E Collins. *Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition.* LNCS, 1975.

[Cou05] P Cousot. Proving program invariance and termination by parametric abstraction, lagrangian relaxation and semidefinite programming. In *Sixth Int. Conf. on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, pages 1–24, Paris, France, LNCS 3385, January 17–19 2005.

[CXYZ07] Yinghua Chen, Bican Xia, Lu Yang, and Naijun Zhan. Generating polynomial invariants with discoverer and qepcad. In *Formal Methods and Hybrid Real-Time Systems*, pages 67–82, 2007.

[Dij76] E W Dijkstra. *A Discipline of Programming.* Prentince-Hall, 1976.

[Fau99] Jean-Charles Faugere. A new efficient algorithm for computing grobner bases (f4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[Flo67] R W Floyd. Assigning meanings to programs. In *Proc. 19th Symp.Applied Mathematics*, pages 19–37, 1967.

[GT06] S. Gulwani and A. Tiwari. Assertion checking over combined abstraction of linear arithmetic and uninterpreted functions. In P. Sestoft, editor, *European Symp. on Programming, ESOP 2006*, volume 3924 of *LNCS*, pages 279–293, 2006.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[JKP06] T. Jebelean, L. Kovacs, and N. Popov. Experimental Program Verification in the Theorema System. *Int. Journal on Software Tools for Technology Transfer (STTT)*, 2006. in press.

[Kap04] D. Kapur. Automatically generating loop invariants using quantifier elimination. *Proc. IMACS Intl. Conf. on Applications of Computer Algebra*, 2004.

[KJ06] L. Kovacs and T. Jebelean. Finding polynomial invariants for imperative loops in the theorema system. In *Proc.of Verify'06 Workshop*, pages 52–67, August 15-16 2006.

[Kov08] Laura Kovacs. Reasoning algebraically about p-solvable loops. In *TACAS 2008: Proc. of the 14th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963, pages 249–264. LNCS, 2008.

[Lan02] Serge Lang. *Algebra*. Springer, January 2002.

[MOS02] Markus Müller-Olm and Helmut Seidl. Polynomial constants are decidable. In *Static Analysis Symposium*, pages 4–19. LNCS, 2002.

[MP95] Zohar Manna and Amir Pnueli. *Temporal verification of reactive systems: safety*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[PC08] André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In *Computer-Aided Verification, CAV 2008, Princeton, USA, Proceedings*, LNCS. Springer, 2008.

[PJ04] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates, 2004.

[RCK07a] E. Rodríguez-Carbonell and D. Kapur. Automatic generation of polynomial invariants of bounded degree using abstract interpretation. *Sci. Comput. Program.*, 64(1):54–75, 2007.

[RCK07b] E. Rodríguez-Carbonell and D. Kapur. Generating all polynomial invariants in simple loops. *J. Symb. Comput.*, 42(4):443–476, 2007.

[RM09] Rachid Rebiha and Arnaldo V. Moura. Automated malware invariant generation. In *6th International Conference on Forensic Computer Science, ICoFSC2009 and ICCYBER2009, "Best paper award"*., 2009.

[RM11a] Rachid Rebiha and Arnaldo V. Moura. Algebraic formal methods for invariant generation. In *PhD. Dissertation*, Faculty of Informatics USI, pages 1–209. University of Lugano Switzerland, January 2011.

[RM11b] Rachid Rebiha and Arnaldo V. Moura. Algebraic formal methods for invariant generation. In *PhD. Dissertation*, Insitute of Computing UNICAMP, pages 1–214. University of Campinas, Sao Paulo Brasil, August 2011.

[RM11c] Rachid Rebiha and Arnaldo V. Moura. Semantic malware resistance using inductive invariants. *International Journal of Forensic Computer Science (IJoFCS0). Best paper award.*, 5, 2011.

[RMM08a] Rachid Rebiha, Nadir Matringe, and Arnaldo V. Moura. Endomorphism for non-trivial semi-algebraic loop invariant generation. Technical Report TR-IC-08-31, Institute of Computing, University of Campinas, November 2008.

[RMM08b] Rachid Rebiha, Nadir Matringe, and Arnaldo V. Moura. Endomorphisms for non-trivial non-linear loop invariant generation. In *5th Int. Conf. Theoretical Aspects of Computing*, pages 425–439. LNCS, 2008.

[RMM10] Rachid Rebiha, Nadir Matringe, and Arnaldo V. Moura. Generatin invariants for non-linear hybrid systems by linear algebraic methods. In *17th Int. Static Analysis Symposium, SAS2010*. LNCS, 2010.

[SA08] S. Gulwani and A. Tiwari. Constraint-based approach for analysis of hybrid systems. In *Proc. of the 14th Int. Conf. on Computer Aided Verification CAV*, 2008.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.

[SSM04a] S. Sankaranarayanan, H. Sipma, and Z. Manna. Constructing invariants for hybrid system. In *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 539–554. Springer, March 2004.

[SSM04b] Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Non-linear loop invariant generation using grobner bases. In *POPL '04: Proc. of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 318–329, New York, NY, USA, 2004. ACM Press.

[SYH96] S. Bensalem, Y. Lakhnech, and H. Saidi. Powerful techniques for the automatic generation of invariants. In Rajeev Alur and Thomas A. Henzinger, editors, *Proc. of the 8th Int. Conf. on Computer Aided Verification CAV*, volume 1102, pages 323–335, NJ, USA, 1996.

[Wei97] Volker Weispfenning. Quantifier elimination for real algebra - the quadratic case and beyond. *Applicable Algebra in Engineering, Communication and Computing*, 8(2):85–101, 1997.

APPENDIX

Here we give more technical details concerning the facts stated in the text. For the ease of reference, we also include original statements.

Theorem 4.1 Consider a transition system corresponding to a loop τ as described in Eq. (4). 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 $Q(L_1(X_1, .., X_n), .., L_n(X_1, .., X_n)) = \lambda Q(X_1, .., X_n)$.

Proof. If $Q(X'_1, .., X'_n) - \lambda Q(X_1, .., X_n)$ belongs to the ideal I generated by the family $(X'_1 - L_1, \ldots, X'_n - L_n)$, then there exists a family $(A_1, \ldots, A_n)$ of polynomials in $\mathbb{R}[X'_1, .., X'_n, X_1, .., X_n]$ such that

$$Q(X'_1, .., X'_n) - \lambda Q(X_1, .., X_n) = (X'_1 - L_1)A_1 + \cdots + (X'_n - L_n)A_n.$$

Letting $X'_i = L_i$, we obtain $Q(L_1(X_1, ..., X_n), .., L_n(X_1, ..., X_n)) = \lambda Q(X_1, ..., X_n)$.

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

Theorem 4.2 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.

Proof. Let Q be a polynomial in $\mathbb{R}_r[X_1,..,X_n]$.

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

Using the definition of an invariant and theorem 4.1, we can see that Q will be a λ-scale invariant if and only if it belong to the eigenspace correspoinding to λ. □

Corollary 4.1 Let M the matrix introduce in this section, departing from its charaterisitcs one could find several decidable classes for the problem of finding a *non-trivial λ-invariant*. For instance one can list the following *decidable* classes:

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

Proof. Suppose M is block triangular with blocks 4×4 or less, then it's characteristic polynomial will a product of polynomials of degree less than four, whose roots can be calculated by Lagrange's 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. Whereas if it is of dimension two or more, than we get polynomials that are non trivial in the eigenspace. Looking at theorem 4.2 to come, we see that it is particularly interesting case. □

Theorem 4.3 A polynomial Q in $\mathbb{R}_r[X_1,..,X_n]$ is an inductive invariant for the affine loop 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 $Q(u_1,\ldots,u_n) = 0$.

Proof. We first consider Theorem 4.2. The initiation step defines on $\mathbb{R}_r[x_1,\ldots,x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1,...,u_n)$. Hence, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1,..,X_n]$ given by the kernel I_u, which is $\{Q \in \mathbb{R}_r[X_1,..,X_n] | Q(u_1,\ldots,u_n) = 0\}$. If we add initial conditions of the form $(x_1(0) = u_1,\ldots,x_n(0) = u_n)$, we are looking for a λ-scale invariant in $\mathbb{R}_r[x_1,\ldots,x_n]$ that belongs to the hyperplane $P(u_1,\ldots,u_n) = 0$, *i.e.*, we are looking for Q in $ker(M - \lambda I) \cap \{P \mid P(u_1,\ldots,u_n) = 0\}$. □

Corollary 4.2 There will be a non-null invariant polynomial for any given initial values if and only if there exists an eigenspace of M with dimension at least 2.

Proof. We take each direction, in turn.

($\Rightarrow$) If 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).Taking any nonzero vector Q in the eigenspace (i.e. a λ-invariant), Q should lie in any hyperplane of initial values,i.e. for every n-tuple $(u_1, \ldots, u_n)$, one would have $Q(u_1, \ldots, u_n) = 0$, i.e $Q = 0$, which is absurd.

($\Leftarrow$) Any eigenspace of M with dimension at least 2 will intersect any space (semi-hyperplan, ...) given by any initial constraints. As any hyperplane is of codimension one in V_r, it must have a nonzero intersection with any subspace of dimension strictly greater than one.

This establishes the result. □

Theorem 5.1 A polynomial Q in $\mathbb{R}[X_1, .., X_n]$ is a T-scale discrete invariant for polynomial-scale consecution with *parametric polynomial* $T \in \mathbb{R}[X_1, ..., X_n]$ for τ if and only if

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

Proof. ($\Rightarrow$) If $Q(X'_1, .., X'_n) - TQ(X_1, .., X_n)$ belongs to the ideal I generated by the family

$$(X'_1 - P_1, \ldots, X'_n - P_n),$$

then there exists a family $(A_1, \ldots, A_n)$ 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 $Q(P_1(X_1, \ldots, X_n), .., P_n(X_1, \ldots, X_n)) = TQ(X_1, \ldots, X_n)$. Then as $Q(X'_1, .., X'_n)$ is equal to $Q(P_1, .., P_n)$ modulo the ideal I, we get $Q(X'_1, .., X'_n) = \lambda Q(X_1, \ldots, X_n)$ modulo I.

This establishes the result. □

Theorem 5.2 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 a T-scale invariant.

Proof. Let Q be a polynomial in $\mathbb{R}[X_1, .., X_n]$. In fact, 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 equivalent terms of matrices

$$((M - L)Q = 0) \Leftrightarrow (Q \in Ker(M - L)),$$

we get the statement of the theorem. □

Theorem 5.3 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 $(0, \ldots, 0, 1)$.

Proof. 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 $\mathbb{R}_r[x_1, \ldots, x_n]$. 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 $V_1(t_1, ..., t_{v(d-1)}), ..., V_s(t_1, ..., t_{v(d-1)})$. From the form of L, this is zero when $(t_1, ..., t_{v(d-1)}) = (0, ..., 0)$. Hence, $M - L$ has its last column equal to zero, giving a common root for these polynomials, corresponding to the constant invariants. □

Theorem 5.4 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 $L \neq 0$ (the 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 correspond to vectors in the intersection.

Proof. Consider Theorem 5.2. The initiation step defines on $\mathbb{R}_r[x_1, \ldots, x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1, ..., u_n)$. Thus, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1, .., X_n]$ given by the kernel I_u, which is $\{Q \in \mathbb{R}_r[X_1, .., X_n] | Q(u_1, \ldots, u_n) = 0\}$. With initial conditions $(x_1(0) = u_1, \ldots, x_n(0) = u_n)$, we are looking for a T-scale differential invariant in $\mathbb{R}_r[x_1, \ldots, x_n]$ that belongs to the hyperplane $P(u_1, \ldots, u_n) = 0$, *i.e.*, we are looking for Q in $ker(M - L) \cap \{P \mid P(u_1, \ldots, u_n) = 0\}$. □

Corollary 5.1 There are non-trivial invariant 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.

Proof.

- ($\Rightarrow$) 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,i.e. for every n-tuple $(u_1, \ldots, u_n)$, one would have $Q(u_1, \ldots, u_n) = 0$,i.e $Q = 0$, which is absurd.
- ($\Leftarrow$) Any intersection between an eigenspace of $M_D - L_T$ with dimension at least 2 will intersect any space (semi-hyperplane, ...) given by any initial constraints.

We get the result. □

Corollary 6.1 Let $\overline{M-L} = U \cdot S \cdot V$ be the singular value decomposition of matrix $\overline{M-L}$ 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.

Proof. 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. □

Theorem 7.1 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

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

Proof.

(⇒) If $Q(X'_1, .., X'_n) - FQ(X_1, .., X_n)$ belongs to the fractional ideal J generated by the family

$$(X'_1 - P_1/Q_1, \ldots, X'_n - P_n/Q_n),$$

then there exists a family $(A_1, \ldots, A_n)$ 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 $Q(\frac{P_1}{Q_1}, .., \frac{P_n}{Q_n}) = \lambda Q(X_1, \ldots, X_n)$.

(⇐) 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. □

Theorem 7.2 Consider M and L as described above. Then, there exists 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 a F-scale discrete invariant.

Proof. 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

$$\Pi^r Q(P_1(X_1,..,X_n),..,P_n(X_1,..,X_n)) = T(X_1,..,X_n)Q(X_1,..,X_n),$$

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 equivalent terms of matrices:

$$((M - L)Q = 0) \Leftrightarrow (Q \in Ker(M - L)),$$

we get the statement of the theorem. □

Theorem 7.3 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 corresponding to vectors in the intersection.

Proof. We first consider Theorem 7.2. The initiation step defines on $\mathbb{R}_r[x_1,\ldots,x_n]$ a linear form on this space, namely, $I_u : P \mapsto P(u_1,...,u_n)$. Hence, initial values correspond to a hyperplane of $\mathbb{R}_r[X_1,..,X_n]$ given by the kernel I_u, which is $\{Q \in \mathbb{R}_r[X_1,..,X_n] | Q(u_1,\ldots,u_n) = 0\}$. With initial conditions $(x_1(0) = u_1,\ldots,x_n(0) = u_n)$, we are looking for a *strong*-scale differential invariant in $\mathbb{R}_r[x_1,\ldots,x_n]$ that belongs to the hyperplane $P(u_1,\ldots,u_n) = 0$, *i.e.*, we are looking for Q in $ker(M - L) \cap \{P \mid P(u_1,\ldots,u_n) = 0\}$. □

Corollary 7.1 We will have a non-trivial invariant for any non-trivial initial value if there exists a matrix L such that the dimension of $Ker(M - L)$ is at least 2.

Proof.

(⇒) 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 $(u_1,\ldots,u_n)$, one would have $Q(u_1,\ldots,u_n) = 0$,i.e $Q = 0$, which is absurd.

(⇐) Any intersection between an eigenspace of M with dimension at least 2 will intersect any space (semi-hyperplane, ...) given by any initial constraints.

And we have the result. □

Theorem 8.1 Let $I = \{I_1,...,I_k\}$ a set of ideals in $\mathbb{R}[X_1,...,X_n]$ such that $I_j = (f^{(j)}{}_1,...,f^{(j)}_{n_j})$ where $j \in [1,k]$. Let's $\otimes(I_1,...,I_k) = \{\delta_1,...,\delta_{n_1 n_2 \ldots n_k}\}$ such that all elements δ_i in $\otimes(I_1,...,I_k)$ are formed by the product of one element from each ideal in I. Assume that all I_js are ideals of invariants for a loop at location l_j described by a transition τ_j. Now, if all l_j describe the same location/program point l, then we have several transitions *looping* at the same point. So we obtain an encoding of possible execution paths of a loop containing *conditional statements*. Then $\otimes(I_1,...,I_k)$ is an ideal of non-trivial non-linear invariants for the entire loop located at l.

Proof. Let $f_1^{(j)}, ..., f_{n_j}^{(j)} \in K[X_1, ..., X_n]$ such that $I_j = (f_1^{(j)}, ..., f_{n_j}^{(j)})$, forall j in $[1, k]$. Let $\beta \in (\otimes(I_1, ..., I_k))$, then there exists $e_1, ..., e_{n_1 n_2 ... n_k} \in K[X_1, .., X_n]$ such that $\beta = e_1\delta_1 + ... + e_{n_1 n_2 ... n_k}\delta_{n_1 n_2 ... n_k}$. Also, by construction of $\otimes(I_1, ..., I_k)$ we know that: $\forall r \in [1, ..., n_1 n_2 ... n_k]$, $\delta_r \in \otimes(I_1, ..., I_k)$. In other words, there is $(\alpha_1^{(r)}, ..., \alpha_k^{(r)}) \in I_1 \times ... \times I_k$ such that $\delta_r = \Pi_{i=0}^{k} \alpha_i^{(r)}$. Then we have $\beta = \sum_{j=1}^{n_1 n_2 ... n_k} [\lambda_j \Pi_{i=1}^{k} \alpha_i^{(j)}]$. Now, for all m in $[1, k]$, if I_m correspond to a pre-computer inductive ideal of invariant associated to one of the transition τ_m at the location l, then $\forall j \in [1, n1n2..nk]$, $\alpha_m^{(j)}(X1, ..., Xn) = 0$. And so $\forall j \in [1, n_1 n_2 ..nk]$, $\Pi_{i=1}^{k} \alpha_i^{(j)} = 0$. Finally we obtain $\beta(X_1, ..., X_n) = 0$ for all m in $[1, n_1 n_2 ..n_k]$. In other words, $\beta(X_1, ..., X_n) = 0$ is an algebraic assertion true at any step of the iteration of the loop for any transition τ_m that could possibly taken. Then $(\beta(X_1, ..., X_n) = 0)$ is an inductive invariant and we can conclude that $(\otimes(I_1, ..., I_k))$ is an ideal of inductive invariant. □