A Sums-of-Squares Extension of Policy Iterations

Assalé Adjé, Pierre-Loïc Garoche, Victor Magron

A Sums-of-Squares Extension of Policy Iterations

Assalé Adjé[1,a] and Pierre-Loïc Garoche[1,a] and Victor Magron[2,b]

[1] Onera, the French Aerospace Lab, France.
Université de Toulouse, F-31400 Toulouse, France.
assale.adje@onera.fr
[2] Circuits and Systems Group, Department of Electrical and Electronic Engineering,
Imperial College London, South Kensington Campus, London SW7 2AZ, UK.
v.magron@imperial.ac.uk

Abstract. In order to address the imprecision often introduced by widening operators, policy iteration based on min-computations amounts to consider the characterization of reachable states of a program as an iterative computation of policies, starting from a post-fixpoint. Computing each policy and the associated invariant relies on a sequence of numerical optimizations. While the early papers rely on LP to address linear properties of linear programs, the current state of the art is still limited to the analysis of linear programs with at most quadratic invariant, relying on Semi-Definite Programming (SDP) solvers to compute the next policy, and LP solvers to solve the selected policy.
We propose here to extend the class of programs considered through the use of Sums-of-Squares (SOS) optimizations. Our approach enables the precise analysis of switched systems with polynomial assigns and guards. The analysis presented has been implemented in Matlab and applied on existing programs, improving both the set of systems analyzable and the precision of analyzed ones.

1 Introduction

A wide set of critical systems software rely on numerical computations: controller systems. Those systems range from aircraft controllers, car engine control, anti-collision systems for aircrafts or UAVs, to nuclear powerplant monitors and medical devices such a pacemakers or insulin pumps.

In all those cases, the software part implements the execution of an endless loop that reads the sensor inputs, updates its internal states and controls actuators. However the analysis of such software is hardly feasible with classical static analysis tools based on linear abstractions. In fact, according to early results in control theory from Lyapunov in the 19th century, a linear system is defined as asymptotically stable iff it satisfies the Lyapunov criteria, the existence of a

[a] The author is supported by the RTRA /STAE Project BRIEFCASE and the ANR ASTRID VORACE Project.
[b] The author is supported by EPSRC (EP/I020457/1) Challenging Engineering Grant.

quadratic invariant. In this view, it is important, in order to support the analysis of these systems, to develop new analysis tools able to support quadratic or richer polynomial invariant.

We are interested here in bounding the set of reachable values of such controllers using sound analysis, that is computing a sound over approximation of reachable states. We are specifically focused on a larger class of programs than linear systems: constrained piecewise polynomial systems. This class of program is represented in Fig. 1: while a condition $cond_0$ is met, depending on another condition $cond_i$ a polynomial update is performed. It is assumed without loss of generality that the n cases of the switch form a partition i.e. $\vee_{1\leq i\leq n} cond_i = \mathbf{true}$ and $\forall i \neq j \in [1,n], cond_i \wedge cond_j = \mathbf{false}$.

To analyze those programs we will rely on policy iterations performed on polynomial templates.

```
init
while (cond0) {
  if (cond1)
    x = poly1(x)
  elseif (cond2)
    x = poly2(x)
  elseif (cond3)
    x = poly3(x);
}
```

Fig. 1. Programs considered.

Related Works

Template abstractions were introduced in [SSM04] as a way to define an abstraction based on an a-priori know vector of templates, i.e. numerical expressions over the program variables. An abstract element is then defined as a vector of reals defining bounds b_i over the templates p_i: $p_i(x_1, \ldots, x_d) \leq b_i$.

Initially templates were used in the classical abstract interpretation setting, computing Kleene fixpoints and the functions p_i were linear. Typically, the value of bounds b_i will be increased during the fixpoint computation until it stabilizes on a postfixpoint.

Later in [GSA+12] the authors proposed to consider richer templates – generalized templates – such as quadratic forms and compute directly the fixpoint of these template-based functions using numerical optimization. When considering specific classes of programs such linear programs, the fixpoint can be computed using a bounded sequence of numerical optimizations. However these methods were bound to linear (LP) or semi-definite (SDP) solvers.

Two dual approaches could be applied. Max-policies [GS07] iterate from initial states and compute *policies* as relaxations through rewriting of an optimization problem (forgetting about rank conditions). Min-policies [GGTZ07,AGG10] rely on duality principle and determine a policy through the computation of a Lagrange multiplier.

The current paper is rooted in this second approach and proposes to enlarge the set of solvers used as well as the class of programs considered:

- address the analysis of constrained piecewise polynomial systems;
- using Sum-of-Square (SOS) based policy iteration, i.e. substituting Lagrange multipliers by SOS multipliers.

The paper is structured as follows: we first characterize the class of programs considered – Constrained Piecewise Discrete-Time Polynomial Systems – and characterize their collecting semantics as a least fixpoint. Then, in Section 3, we recall definitions of generalized templates, their expression as an abstract domain and the definition of the abstract transfer function. Section 4 proposes an abstraction of the transfer function using a SOS relaxation, while Section 5 relies on this abstraction to perform policy iteration. Finally Section 6 presents experiments.

2 Constrained Piecewise Discrete-Time Polynomial Systems: Definition and Collecting Semantics

In this paper, we are interested in proving automatically that the set of all the possible trajectories of a dynamical system is bounded. We consider the special class of discrete-time dynamical systems: (i) their dynamic law is a piecewise polynomial function, and (ii) the state variable is constrained to live in some given basic semi-algebraic set[c]. Note that f is a piecewise polynomial function with respect to a given partition, meaning that if we restrict f to be an element of the partition then f is a polynomial function. We recall that a set is a basic semi-algebraic set if and only if it can be represented as a conjunction of strict or weak polynomial inequalities.

Let $\mathcal{I}$ be a finite set of partition labels. Let $\mathcal{X} = \{X^i, i \in \mathcal{I}\} \subseteq \mathbb{R}^d$ be a partition, that is a given family of basic semi-algebraic sets satisfying Equation (1):

$$\bigcup_{i \in \mathcal{I}} X^i = \mathbb{R}^d,\ \forall i, j \in \mathcal{I},\ i \neq j, X^i \cap X^j \neq \emptyset\ . \tag{1}$$

By definition of basic semi-algebraic sets, for all $i \in \mathcal{I}$, there exists a family of n_i polynomials $\{r^i_j, j \in [n_i]\}$ such that:

$$X^i = \left\{x \in \mathbb{R}^d \mid r^i_j(x) \bowtie 0\ \forall\, j \in [n_i]\right\}\ . \tag{2}$$

where $\bowtie$ is either $<$ or $\leq$ and $[n_i]$ denotes the set $\{1, \ldots, n_i\}$.

Now let $T : \mathbb{R}^d \mapsto \mathbb{R}^d$ be a piecewise polynomial function w.r.t. to the partition $\mathcal{X}$. By definition, there exists a family of polynomials $\{T_i : \mathbb{R}^d \mapsto \mathbb{R}^d, i \in \mathcal{I}\}$ such that for all $i \in \mathcal{I}$:

$$T(x) = T^i(x),\ \forall x \in X^i\ . \tag{3}$$

Finally, let X^{in} and X^0 be two basic semi-algebraic sets of $\mathbb{R}^d$, X^{in} supposed to be compact, i.e. closed and bounded. The two sets can be represented as in Equation (2) using their respective family of n_{in} and n_0 polynomials:

$$X^{\text{in}} = \left\{x \in \mathbb{R}^d \mid r^{\text{in}}_j(x) \bowtie 0\ \forall\, j \in [n_{\text{in}}]\right\} \text{ and } X^0 = \left\{x \in \mathbb{R}^d \mid r^0_j(x) \bowtie 0\ \forall\, j \in [n_0]\right\}$$

where for all $j \in [n_{\text{in}}]$, $r^{\text{in}}_j : \mathbb{R}^d \mapsto \mathbb{R}$ and for all $k \in [n_0]$, $r^0_k : \mathbb{R}^d \mapsto \mathbb{R}$ are polynomials.

[c] This does not entail boundedness of variable values such as membership of the sublevel set $\{x \in \mathbb{R}^d \mid 1 - \|x\|_2^2 \leq 0\}$.

To sum up, we define a *constrained piecewise discrete-time dynamical system* (PPS for short) as a system verifying the following dynamic:

$$x_0 \in X^{\text{in}}, \text{ and } \forall\, k \in \mathbb{N}, \text{if } x_k \in X^0,\ x_{k+1} = T(x_k)\,. \tag{4}$$

We are only interested in the iterates of the PPS that live in X^0: either the infinite traces $x_0 \cdot x_i \cdots$ where $x_0 \in X^{\text{in}}$ and $\forall\, i \in \mathbb{N}^*, x_i \in X^0$ or the finite traces $x_0 \cdots x_n$ where $x_0 \in X^{\text{in}}$ and $\forall\, i \in [n], x_i \in X^0, x_n \notin X^0$. Intuitively, this system encodes the example program of Figure 1.

From Equation (1), for all $k \in \mathbb{N}^*$ there exists a unique $i \in \mathcal{I}$ such that $x_k \in X^i$. Let us now give a formal definition of PPS.

Definition 1 (PPS). *A constrained piecewise discrete-time dynamical system (PPS) is the quadruple $(X^{\text{in}}, X^0, \mathcal{X}, \mathcal{L})$ with:*

- *$X^{\text{in}} \subseteq \mathbb{R}^d$ is the compact basic semialgebraic set of the possible initial conditions;*
- *$X^0 \subseteq \mathbb{R}^d$ is the set of global constraints to be satisfied;*
- *$\mathcal{X} := \{X^i, i \in \mathcal{I}\}$ is a partition as defined in Equation (1);*
- *$\mathcal{L} := \{T^i, i \in \mathcal{I}\}$ is the family of the functions from $\mathbb{R}^d$ to $\mathbb{R}^d$, w.r.t. the partition $\mathcal{X}$ satisfying Equation (4).*

Example 1 (Running example). We consider the following running example. It corresponds to a slightly modified version of [AJ13, Example 3]. We have added semi-algebraic sets to represent conditions under which we can activate a polynomial update. $(X^{\text{in}}, X^0, \{X^1, X^2\}, \{T^1, T^2\})$, where:

$$\begin{array}{l} X^{\text{in}} = [-1,1] \times [-1,1] \\ X^0 = \mathbb{R}^d \end{array} \quad \text{and} \quad \begin{cases} X^1 = \{x \in \mathbb{R}^2 \mid -x_1^2 + 0.5x_2 + 0.5 \leq 0\} \\ X^2 = \{x \in \mathbb{R}^2 \mid x_1^2 - 0.5x_2 - 0.5 < 0\} \end{cases}$$

and the functions relative to the partition $\{X^1, X^2\}$ are:

$$T^1 = \begin{pmatrix} 0.687x_1 + 0.558x_2 - 0.0001x_1x_2 \\ -0.292x_1 + 0.773x_2 \end{pmatrix}$$

and

$$T^2 = \begin{pmatrix} 0.369x_1 + 0.532x_2 - 0.0001x_1^2 \\ -1.27x_1 + 0.12x_2 - 0.0001x_1x_2 \end{pmatrix}$$

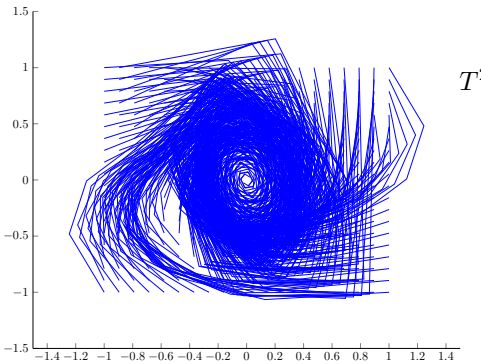

Fig. 2. Running example simulation

We recall that our objective is to prove automatically that the set of the possible trajectories of the system is bounded. Let us define the set of possible trajectories also known as the reachable values set or the collecting semantics of the system:

$$\mathfrak{R} = \bigcup_{k \in \mathbb{N}} T^k_{|_{X^0}}(X^{\text{in}}) \tag{5}$$

where $T_{|_{X^0}}$ is the restriction of T on X^0 and $T_{|_{X^0}}$ is not defined outside X^0. To prove this boundedness property, we can compute $\mathfrak{R}$ and do some analysis to prove that $\mathfrak{R}$ is bounded. Nevertheless, the computation of $\mathfrak{R}$ cannot be done in general and we have to compute instead an over-approximation of it. Then we show that the over-approximation is bounded and we exhibit the smallest computable bound.

The usual abstract interpretation methodology to characterize and construct this over-approximation relies on the representation of $\mathfrak{R}$ as the smallest fixed point of a monotone map over a complete lattice. Indeed, we can remark that we can reformulate $\mathfrak{R}$ as follows: $\mathfrak{R} = T(\mathfrak{R} \cap X^0) \cup X^{\text{in}} = \bigcup_{i \in \mathcal{I}} T^i(\mathfrak{R} \cap X^0 \cap X^i) \cup X^{\text{in}}$ and thus introducing the function $F : \wp(\mathbb{R}^d) \mapsto \wp(\mathbb{R}^d)$ defined for all $C \in \wp(\mathbb{R}^d)$ by: $F(C) = T(C \cap X^0) \cup X^{\text{in}} = \bigcup_{i \in \mathcal{I}} T^i(C \cap X^0 \cap X^i) \cup X^{\text{in}}$ then, $\mathfrak{R}$ is the smallest fixed point of F and from Tarski's theorem, since F is monotonic and $\wp(\mathbb{R}^d)$ a complete lattice, $\mathfrak{R} = \min\{C \in \wp(\mathbb{R}^d) \mid F(C) \subseteq C\}$. Finally to compute an over-approximation of $\mathfrak{R}$ it suffices to compute a set C such that $F(C) \subseteq C$. A set C which satisfies $F(C) \subseteq C$ is called an *inductive invariant*.

The rest of the paper addresses the computation of a sound over-approximation of $\mathfrak{R}$ using its definition as the least fixpoint of F.

3 Templates Abstract Domains

Working directly with sets is difficult. We propose to restrict the class of inductive invariant considered to some basic semi-algebraic sets using template abstractions. Recall that a multivariate polynomial $\mathbb{R}^d \mapsto \mathbb{R}$ of degree k can be expressed as $\sum_{|\alpha| \leq k, |\alpha| > 0} c_\alpha x^\alpha + c_0$ where α is vector of integers of size d, $x^\alpha = (x_i^{\alpha_i})_{i \in [d]}$ and $|\alpha|$ denotes the sum of the coordinates of α. We can interpret $\sum_{|\alpha| \leq k, |\alpha| > 0} c_\alpha x^\alpha$ as the homogeneous part of the polynomial and c_0 the constant part. The class of basic semi-algebraic which we will consider is the class of sets sharing the same fixed a priori homogeneous part but differ from the constant part. Since we want to prove that $\mathfrak{R}$ is bounded, the basic semi-algebraic inductive invariant has to be bounded. Furthermore, since we restrict the class of inductive invariant to the basic semi-algebraic sets sharing a same fixed a priori homogeneous part, the image by F of these basic semi-algebraic sets has to be in the same representation.

This method is called the templates abstraction [AGG10] and we specialize the method here to the semi-algebraic set case.

3.1 Generalized Templates Abstract Domains

The concept of generalized templates was introduced in [AGG10,AGG11]. Let $\mathbf{F}\left(\mathbb{R}^d, \mathbb{R}\right)$ stands for the set of functions from $\mathbb{R}^d$ to $\mathbb{R}$.

Definition 2 (Generalized templates). *A generalized template p is a function from $\mathbb{R}^d$ to $\mathbb{R}$ over the vector of variables $(x_1, \ldots, x_d)$.*

Templates can be viewed as implicit functional relations on variables to prove certain properties on the analyzed program. We denote by $\mathbb{P}$ the set of templates.

First, we suppose that $\mathbb{P}$ is given by some oracle and say that $\mathbb{P}$ forms a template basis. Here, we recall the required background about generalized templates (see [AGG10,AGG11] for more details).

Basic notions. We replace the classical concrete semantics using sub-level sets i.e. we have a functional representation of numerical invariant through the functions of $\mathbb{P}$. An invariant is determined as the intersection of sub-level sets. The problem is thus reduced to find optimal level sets on each template p. Let $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ stands for the set of functions from $\mathbb{P}$ to $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$, that is associating bounds to templates.

Definition 3 ($\mathbb{P}$-sublevel sets). *For each $w \in \mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$, we associate the $\mathbb{P}$-sublevel set $w^\star \subseteq \mathbb{R}^d$ given by:*

$$w^\star = \{x \in \mathbb{R}^d \mid p(x) \leq w(p),\ \forall p \in \mathbb{P}\} = \bigcap_{p \in \mathbb{P}} \{x \in \mathbb{R}^d \mid p(x) \leq w(p)\}\,.$$

In convex analysis, a closed convex set can be represented by its support function i.e. the supremum of linear forms on the set (e.g. [Roc96, § 13]). Here, we use the generalization by Moreau [Mor70] (see also [Rub00,Sin97]) which consists in replacing the linear forms by the functions $p \in \mathbb{P}$.

Definition 4 ($\mathbb{P}$-support functions). *For each $X \subseteq \mathbb{R}^d$, we associate the abstract support function denoted by $X^\dagger : \mathbb{P} \mapsto \overline{\mathbb{R}}$ and defined by:*

$$X^\dagger(p) = \sup_{x \in X} p(x)\,.$$

Let C and D be two ordered sets equipped respectively by the order $\leq_C$ and $\leq_D$. Let ψ be a map from C to D and φ be a map from D to C. We say that the pair (ψ, φ) defines a Galois connection between C and D if and only if ψ and φ are monotonic and the equivalence $\psi(c) \leq_D d \iff \varphi(d) \leq_C c$ holds for all $c \in C$ and all $d \in D$.

We equip $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ with the partial order $\leq_{\mathbf{F}}$ of real-valued functions i.e. $w \leq_{\mathbf{F}} v \iff w(p) \leq v(p)\ \forall p \in \mathbb{P}$. The set $\wp(\mathbb{R}^d)$ is equipped with the inclusion order $\sqsubseteq$.

Proposition 1. *The pair of maps $w \mapsto w^\star$ and $X \mapsto X^\dagger$ defines a Galois connection between $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ and the set of subsets of $\mathbb{R}^d$.*

In the terminology of abstract interpretation, $(\cdot)^\dagger$ is the abstraction function, and $(\cdot)^\star$ is the concretization function. The Galois connection result provides the correctness of the semantics. We also remind the following property:

$$(((w^\star)^\dagger)^\star = w^\star\,, \qquad ((X^\dagger)^\star)^\dagger = X^\dagger\,. \tag{6}$$

The lattices of $\mathbb{P}$-convex sets and $\mathbb{P}$-convex functions. When fixing the set of templates, we can characterize such lattice structure. We are now interested in the Moore family of $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ generated by the closure of $\mathbb{R}^d$ through $\dagger$, and in its image by $\star$. We denote these sets as $\mathbb{P}$-convex.

Definition 5 ($\mathbb{P}$-convexity). *Let $w \in \mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$, we say that w is a $\mathbb{P}$-convex function if $w = (w^\star)^\dagger$. A set $X \subseteq \mathbb{R}^d$ is a $\mathbb{P}$-convex set if $X = (X^\dagger)^\star$. We respectively denote by $\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ and $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ the set of $\mathbb{P}$-convex functions of $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ and the set of $\mathbb{P}$-convex sets of $\mathbb{R}^d$.*

The family of functions $\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ is ordered by the partial order of real-valued functions. The family of sets $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ is ordered by the inclusion order. Galois connection allows to construct lattice operations on $\mathbb{P}$-convex elements.

Definition 6 (The meet and join). *Let v and w be in $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$. We denote by $\inf(v, w)$ and $\sup(v, w)$ the functions defined respectively by, $p \mapsto \inf(v(p), w(p))$ and $p \mapsto \sup(v(p), w(p))$. We equip $\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ with the join operator $v \vee w = \sup(v, w)$ and the meet operator $v \wedge w = (\inf(v, w)^\star)^\dagger$. Similarly, we equip $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ with the join operator $X \sqcup Y = ((X \cup Y)^\dagger)^\star$ and the meet operator $X \sqcap Y = X \cap Y$.*

The next theorem follows readily from the fact that the pair of $v \mapsto v^\star$ and $C \mapsto C^\dagger$ defines a Galois connection (see e.g. [DP02, § 7.27]).

Theorem 1. *The complete lattices $(\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}}), \wedge, \vee)$ and $(\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d), \sqcap, \sqcup)$ are isomorphic.*

3.2 Abstract Transfer Function using Polynomial Templates and Inductive Sublevel Sets Invariant

Let us assume now that $\mathbb{P}$ is a given family of polynomials. This family determines the family of basic semi-algebraic sets which share the same homogeneous part. We consider $\mathbb{P}$-sublevel sets $w^\star$, for $w \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ and we construct the mapping which associates to $w^\star$ an element of $\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$. We are looking for a function $F^\sharp$ such that the image of its inductive invariant by $\star$ are also inductive invariant for F.

In abstract interpretation [CC77], to ensure the latter property we use a Galois connection between $\wp(\mathbb{R}^d)$ and the abstract domain. Here the abstract domain is the templates abstract domain w.r.t. $\mathbb{P}$ and we use the pair of maps $(w \mapsto w^\star, X \mapsto X^\dagger)$ as Galois connection see Proposition 6.

It is now possible to define the abstract transformer $F^\sharp$, endomorphism of $\mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$. We recall that, in presence of a Galois connection (α, γ), the best abstraction [CC77] of a function f is defined as $\alpha \circ f \circ \gamma$.

Here, $F^\sharp(w) = (F(w^\star))^\dagger$ and using the definition of F and the semi-algebraic characterizations of X^0 and X^i, we have for all $w \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ and for all $p \in \mathbb{P}$:

$$
\begin{aligned}
&\left(F^\sharp(w)\right)(p) \\
&\quad = \sup_{y \in \bigcup_i T^i(w^\star \cap X^0 \cap X^i) \cup X^{\mathrm{in}}} p(y) && \text{(by definition of } \dagger \text{ and } F\text{)} \\
&\quad = \sup \left\{ \sup_{y \in \bigcup_i T^i(w^\star \cap X^0 \cap X^i)} p(y), X^{\mathrm{in}^\dagger}(p) \right\} && \text{(by definition of } \cup\text{)}
\end{aligned}
$$

$$= \sup\left\{ \sup_{i\in\mathcal{I}} \sup_{y\in T^i(w^\star\cap X^0\cap X^i)} p(y), {X^{\text{in}}}^\dagger(p) \right\} \qquad \text{(by definition of } \cup\text{)}$$

$$= \sup\left\{ \sup_{i\in\mathcal{I}} \sup_{\substack{x\in w^\star \\ r^i_j(x)\leq 0,\ \forall\, j\in[n_i] \\ r^0_k(x)\leq 0,\ \forall\, k\in[n_0]}} p(T^i(x)), {X^{\text{in}}}^\dagger(p) \right\} \qquad \text{(by definition (2))}$$

$$= \sup\left\{ \sup_{i\in\mathcal{I}} \sup_{\substack{q(x)\leq w(q),\ \forall q\in\mathbb{P} \\ r^i_j(x)\leq 0,\ \forall\, j\in[n_i] \\ r^0_k(x)\leq 0,\ \forall\, k\in[n_0]}} p(T^i(x)), {X^{\text{in}}}^\dagger(p) \right\} \qquad \text{(by definition of } \star\text{)}$$

Let us introduce for each $i \in \mathcal{I}$ the map $F_i^\sharp$ defined for all $w \in \text{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$ and $p \in \mathbb{P}$ by: $\big(F_i^\sharp(w)\big)(p) = \sup_{\substack{q(x)\leq w(q),\ \forall q\in\mathbb{P} \\ r^i_j(x)\leq 0,\ \forall\, j\in[n_i] \\ r^0_k(x)\leq 0,\ \forall\, k\in[n_0]}} p(T^i(x))$

Recall that p and T^i are polynomials. Evaluating $\big(F_i^\sharp(w)\big)(p)$ amounts then to solve a polynomial maximization problem.

4 Relaxed semantics SOS based

In this section, we introduce the relaxed functional on which we will compute a fixpoint in order to define an over-approximation of the reachable values set $\mathfrak{R}$. The relaxed functional is constructed from a Lagrange relaxation type of maximization problems involved in the evaluation of $F^\sharp$ and sums-of-squares reinforcement of polynomial non-negativity constraints. First, we give the useful notions of sums-of-squares and their use in a polynomial optimization context. The interested reader is referred to [Las09] for further information.

4.1 Sum-of-Square (SOS) Programming

Let $\mathbb{R}[x]_k$ be the set of real polynomials of degree at most k defined over variables x. We denote by $\mathbb{R}[x]$ the set of all polynomials over the variable x. A polynomial p is said to be positive if and only if $p(x) \geq 0$ for all $x \in \mathbb{R}^d$. Then to check whether a polynomial is positive is reduced to test the positivity of an infinite number of reals.

We say that $p \in \mathbb{R}[x]_k$ admits a sum-of-squares decomposition if and only if there exists $g_1, \ldots, g_l$ polynomials such that $p = \sum_{i=1}^{l} g_i^2$. Note that if p admits a sum-of-squares decomposition then the degree of p is even. We denote by $\Sigma[x]_{2m}$ the set of a sum-of-squares decomposable polynomials of degree $2m$ and by $\Sigma[x]$ the set of all sum-of-squares decomposable polynomials. A polynomial $p \in \mathbb{R}[x]_{2m}$ belongs to $\Sigma[x]_{2m}$ if and only if there exists a *semi-definite positive matrix* Q such that for all $x \in \mathbb{R}^d$: $p(x) = v_m(x) Q v_m(x)^\intercal$ where $v_m(x)$ is the vector of all monomials of degree at most m that is $v_m(x) = (1, x_1, x_2, \ldots, x_d, \ldots, x_1^m, \ldots, x_d^m)$. The size of the vector $v_m(x)$ is $\binom{d+m}{d}$ and thus Q is of size $\binom{d+m}{d} \times \binom{d+m}{d}$.

It is obvious that all $p \in \Sigma[x]$ are positive polynomials. To check whether a given polynomial $p \in \mathbb{R}[x]_{2m}$ belongs to $\Sigma[x]$ can be done by solving a semi-definite feasibility program.

The SOS reinforcement of polynomial optimization problems consists in restricting polynomial positivity by being an element of $\Sigma[x]$. In case of polynomial maximization problems, the SOS reinforcement induces a computation of an upper bound on the real optimal value. For example let us consider an unconstrained maximization problem and let $p \in \mathbb{R}[x]$. Applying SOS reinforcement, we obtain:

$$\sup\{p(x),\ x \in \mathbb{R}^d\} = \inf\{\eta \mid \eta - p(x) \geq 0\} \leq \inf\{\eta \mid \eta - p(x) \in \Sigma[x]\}\ . \quad (7)$$

Now, let $p, q \in \mathbb{R}[x]$ and consider the constrained optimization problem: $\sup\{p(x) | q(x) \leq 0\}$. Let $\lambda \in \Sigma[x]$, then: $\sup_{q(x)\leq 0} \quad p(x) \leq \sup_{x\in\mathbb{R}^d} \quad p(x) - \lambda(x) \cdot q(x)$. Indeed, suppose $q(x) \leq 0$, then $-\lambda(x)q(x) \geq 0$ and $p(x) \leq p(x) - \lambda(x)q(x) \geq 0$. Finally taking the supremum over $\{x \in \mathbb{R}^d \mid q(x) \leq 0\}$ provides the inequality. Since $\sup\{p(x) - \lambda(x) \cdot q(x),\ x \in \mathbb{R}^d\}$ is an unconstrained polynomial maximization problem then we can apply a SOS reinforcement and we obtain:

$$\sup_{q(x)\leq 0} \quad p(x) \leq \sup_{x\in\mathbb{R}^d} \quad p(x) - \lambda(x) \cdot q(x) \leq \inf\{\eta \mid \eta - p - \lambda q \in \Sigma[x]\}$$

Finally, note that this latter inequality is valid whatever $\lambda \in \Sigma[x]$ and so we can take the infimum over $\lambda \in \Sigma[x]$ which leads to

$$\sup_{q(x)\leq 0} \quad p(x) \leq \inf_{\lambda\in\Sigma[x]} \sup_{x\in\mathbb{R}^d} \quad p(x) - \lambda(x) \cdot q(x) \leq \inf_{\substack{\eta - p - \lambda q \in \Sigma[x] \\ \lambda \in \Sigma[x]}} \quad \eta \quad (8)$$

Note that since positive scalars can be viewed as sum-of-squares polynomials, we can restrict λ to be a positive scalar. In presence of several constraints, we assign to each constraint an element $\sigma \in \Sigma[x]$, and we consider the product of σ with the associated constraint and then the sum of all products. This sum is finally added to the objective function.

4.2 Relaxed semantics

The computation of $F^\sharp$ as a polynomial maximization problem cannot be directly performed using numerical solvers. We use the SOS reinforcement mechanisms described above to relax the computation and characterize an abstraction of $F^\sharp$. Let us recall the definition of $F_i^\sharp$ and let $w \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{P} \mapsto \overline{\mathbb{R}})$:

$$\begin{aligned}
\big(F_i^\sharp(w)\big)(p) &= \sup_{\substack{q(x)\leq w(q),\ \forall q\in\mathbb{P} \\ r_j^i(x)\leq 0,\ \forall\, j\in[n_i] \\ r_k^0(x)\leq 0,\ \forall\, k\in[n_0]}} p(T^i(x)) \\
&\leq \inf_{\substack{\lambda\in\mathbf{F}(\mathbb{P},\mathbb{R}_+) \\ \mu\in\Sigma[x]^{n_i},\gamma\in\Sigma[x]^{n_0}}} \sup_{x\in\mathbb{R}^d} p(T^i(x)) + \textstyle\sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)(w(q) - q(x)) \\
&\qquad\qquad - \textstyle\sum_{\mathrm{l}=1}^{\mathrm{n^i}} \mu_l(x) r_l^i(x) - \sum_{\mathrm{l}=1}^{\mathrm{n^0}} \gamma_l(x) r_l^0(x) \\
&\leq \inf_{\lambda,\mu,\gamma,\eta} \eta \\
&\qquad \text{s.t.} \begin{cases} \eta - p\circ T^i - \displaystyle\sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)(w(q) - q) + \sum_{\mathrm{l}=1}^{\mathrm{n^i}} \mu_l r_l^i + \sum_{\mathrm{l}=1}^{\mathrm{n^0}} \gamma_l r_l^0 \in \Sigma[x] \\ \text{where } \lambda \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right),\ \mu \in \Sigma[x]^{n_i},\ \gamma \in \Sigma[x]^{n_0},\ \eta \in \mathbb{R} \end{cases} \\
&\qquad\qquad \text{(using a SOS reinforcement to remove the sup)}
\end{aligned}$$

In order to simplify the notations, let us write $\sum_{l=1}^{n^i} \mu_l r_l^i$ (resp. $\sum_{l=1}^{n^0} \gamma_l r_l^0$) as $\langle \mu, r^i \rangle$ ($\langle \gamma, r^0 \rangle$). Finally, we introduce $\big(F_i^{\mathcal{R}}(w)\big)(p)$, the over-approximation of $\big(F_i^{\sharp}(w)\big)(p)$:

$$\big(F_i^{\mathcal{R}}(w)\big)(p) = \inf_{\lambda,\mu,\gamma,\eta} \eta \quad \text{s. t.} \begin{cases} \eta - p \circ T^i - \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q) + \langle \mu, r^i \rangle + \langle \gamma, r^0 \rangle \in \Sigma[x] \\ \text{where } \lambda \in \mathbf{F}(\mathbb{P}, \mathbb{R}_+),\ \mu \in \Sigma[x]^{n_i},\ \gamma \in \Sigma[x]^{n_0},\ \eta \in \mathbb{R} \end{cases}$$

Next, we will need to extend $F_i^{\sharp}$ to the whole space $\mathbf{F}(\mathbb{P}, \mathbb{R})$ and we will restrict the templates bound to finite valued functions on templates.

The computation of $F^{\sharp}$ needs the computation of $X^{\mathrm{in}\dagger}$ that is, by definition, equal to $\sup\{p(x), x \in X^{\mathrm{in}}\}$. Since X^{in} is a basic semi-algebraic and the templates p are polynomials then the evaluation of $X^{\mathrm{in}\dagger}$ is reduced to solve a polynomial maximization problem. We can use SOS reinforcement described above to over-approximate and thus we define $X^{\mathrm{in}\mathcal{R}}$ the relaxed version of $X^{\mathrm{in}\dagger}$:

$$X^{\mathrm{in}\mathcal{R}}(p) := \inf\{\eta \mid \eta - p + \sum_{j=1}^{n_{\mathrm{in}}} \nu_j^{\mathrm{in}} r_j^{\mathrm{in}} \in \Sigma[x],\ \nu^{\mathrm{in}} \in \Sigma[x]^{n_{\mathrm{in}}}\}$$

Note that since X^{in} is a nonempty compact basic semi-algebraic, then $X^{\mathrm{in}\mathcal{R}}(p)$ is finite valued i.e. cannot take the value $+\infty$ neither $-\infty$ [Las09, Th. 2.15].

Finally, we define the relaxed functional $F^{\mathcal{R}}$ for all $w \in \mathbf{F}(\mathbb{P}, \mathbb{R})$ for all $p \in \mathbb{P}$ as follows:

$$\big(F^{\mathcal{R}}(w)\big)(p) = \sup\left\{\sup_{i \in \mathcal{I}} \big(F_i^{\mathcal{R}}(w)\big)(p), X^{\mathrm{in}\mathcal{R}}(p)\right\} \tag{9}$$

By construction, the relaxed function $F^{\mathcal{R}}$ provides a safe over-approximation of the abstract semantics $F^{\sharp}$. Furthermore, for all $i \in \mathcal{I}$, the evaluation of $F_i^{\mathcal{R}}$ can be done using semi-definite programming, since it is reduced to solve a minimization problem with a linear objective function and linear combination of polynomials constrained to be sum-of-squares.

By construction, we have readily the following proposition:

Proposition 2 (Safety). *The following statements hold: (I) For all $p \in \mathbb{P}$, $X^{\mathrm{in}\dagger}(p) \leq X^{\mathrm{in}\mathcal{R}}(p)$; (II) For all $i \in \mathcal{I}$, for all $w \in \mathbf{F}(\mathbb{P}, \mathbb{R})$ and for all $p \in \mathbb{P}$: $\big(F_i^{\sharp}(w)\big)(p) \leq \big(F_i^{\mathcal{R}}(w)\big)(p)$; (III) For all $w \in \mathbf{F}(\mathbb{P}, \mathbb{R})$ and for all $p \in \mathbb{P}$, $\big(F^{\sharp}(w)\big)(p) \leq \big(F^{\mathcal{R}}(w)\big)(p)$.*

An important property that we will use to prove some results on policy iteration algorithm is the monotonicity of the relaxed functional.

Proposition 3 (Monotonicity). *The following statements hold: (I) For all $i \in \mathcal{I}$, $w \mapsto \big(F_i^{\mathcal{R}}(w)\big)$ is monotonic on $\mathbf{F}(\mathbb{P}, \mathbb{R})$; (II) The map $w \mapsto F^{\mathcal{R}}(w)$ is monotonic on $\mathbf{F}(\mathbb{P}, \mathbb{R})$.*

5 Policy Iteration in Polynomial Templates Abstract Domains

We are interested in computing the least fixpoint $\mathfrak{R}^{\mathcal{R}}$ of $F^{\mathcal{R}}$, over-approximation of $\mathfrak{R}$, least fixpoint of F. As for the definition of $\mathfrak{R}$, it can be reformulated using Tarski theorem as the minimal post-fixpoint: $\mathfrak{R}^{\mathcal{R}} = \min\{w \in \mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right) | F^{\mathcal{R}}(w) \leq_{\mathbf{F}} w\}$. The idea behind policy iteration is to compute $\mathfrak{R}^{\mathcal{R}}$ using successive iterations which are composed of a vector bound computation using linear programming and the determination of a new policy when a fixpoint is not reached. Policy iteration navigates in the set of postfixpoints of $F^{\mathcal{R}}$ and needs to start from a postfixpoint w^0 supposed given. It acts like a narrowing operator and can be interrupted as any time. For further information on policy iteration, the interested reader can consult [CGG+05,GGTZ07].

5.1 Policies

Policy iteration can be used to compute a fixpoint of a monotone self-map defined as an infimum of a family of affine monotone self-maps. In this paper, we propose to design a policy iteration algorithm to compute a fixpoint of $F^{\mathcal{R}}$. First, we remark that $F^{\mathcal{R}}$ is not directly written as an infimum but for all $i \in \mathcal{I}$, $F_i^{\mathcal{R}}$ is and so for all $i \in \mathcal{I}$, we apply the concept of policies to $F_i^{\mathcal{R}}$.

Policy iteration needs a *selection property*, that is when a vector is given, there exists a policy which achieves the infimum. In our context since we apply the concept of policies to $F_i^{\mathcal{R}}$, it means that the minimization problem involved in the computation of $F_i^{\mathcal{R}}$ has an optimal solution. For $w \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}\right)$ and $p \in \mathbb{P}$, an optimal solution, in this case, is a vector $(\lambda, \mu, \gamma, g) \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right) \times \Sigma[x]^{n_i} \times \Sigma[x]^{n_0} \times \mathbb{R}[x]^l$ such that:

$$\left(F_i^{\mathcal{R}}(w)\right)(p) = p \circ T^i + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q) - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \sum_{j=1}^{l} g_j^2 \ . \quad (10)$$

We warn the reader that in Equation (10), $\left(F_i^{\mathcal{R}}(w)\right)(p)$ is a scalar whereas $p \circ T^i + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q) - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \sum_{j=1}^{l} g_j^2$ is a priori a polynomial. The equality in Equation (10) means that actually $p \circ T^i + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q) - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \sum_{j=1}^{l} g_j^2$ is a constant polynomial i.e. all coefficients associated to a monomial of degree greater than 1 are zero.

We denote by $\mathrm{Sol}(w, i, p)$ the set of $(\lambda, \mu, \gamma, g) \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right) \times \Sigma[x]^{n_i} \times \Sigma[x]^{n_0} \times \mathbb{R}[x]^l$ such that Equation (10) holds. If $\mathrm{Sol}(w, i, p) = \emptyset$, since policy iteration algorithm can be stopped at any step and still provides a sound over-approximation, we stop the iteration. Then, we define FS as the set of$w \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}\right)$ such that for all $i \in \mathcal{I}$, for all $p \in \mathbb{P}$, $\mathrm{Sol}(w, i, p) \neq \emptyset$. Finally, we can define a policy as a map which selects for all $w \in \mathrm{FS}$, for all $i \in \mathcal{I}$ and for all $p \in \mathbb{P}$ a vector of $\mathrm{Sol}(w, i, p)$. More formally, we have the following definition:

Definition 7 (Policies in the policy iteration SOS based setting). *A policy is a map* $\pi : \mathrm{FS} \mapsto ((\mathcal{I} \times \mathbb{P}) \mapsto \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right) \times \Sigma[x]^{n_i} \times \Sigma[x]^{n_0} \times \mathbb{R}[x]^l)$ *such that:* $\forall\, w \in \mathrm{FS}$, $\forall\, i \in \mathcal{I}$, $\forall\, p \in \mathbb{P}$, $\pi(w)(i, p) \in \mathrm{Sol}(w, i, p)$.

We denote by Π the set of policies and for $\pi \in \Pi$, we write π_λ the map from FS to $(\mathcal{I} \times \mathbb{P}) \mapsto \mathbf{F}(\mathbb{P}, \mathbb{R}_+)$which associates to $w \in \text{FS}$ and $(i,p) \in \mathcal{I} \times \mathbb{P}$ the first coordinate of $\pi(w)(i,p)$ i.e. if $\pi(w)(i,p) = (\lambda, \mu, \gamma, g)$ then $\pi_\lambda(w)(i,p) = \lambda$.

As said before, policy iteration exploits the linearity of maps when a policy is fixed. We have to explicit the affine maps we will use in a policy iteration step. Let $\pi \in \Pi$, $w \in \text{FS}$, $i \in \mathcal{I}$ and $p \in \mathbb{P}$ and let $\lambda = \pi_\lambda(w)(i,p)$, we define the map $\varphi^\lambda_{w,i,p} : \mathbf{F}(\mathbb{P}, \mathbb{R}) \mapsto \mathbb{R}$ as follows:

$$v \mapsto \varphi^\lambda_{w,i,p}(v) = \sum_{q \in \mathbb{P}} \lambda(q) v(q) + \left(F^{\mathcal{R}}_i(w)\right)(p) - \sum_{q \in \mathbb{P}} \lambda(q) w(q) \tag{11}$$

Then, for $\pi \in \Pi$, we define for all $w \in \text{FS}$, $\Phi^{\pi(w)}_w$ the map from $\mathbf{F}(\mathbb{P}, \mathbb{R}) \mapsto \mathbf{F}(\mathbb{P}, \mathbb{R})$ for all $v \in \mathbf{F}(\mathbb{P}, \mathbb{R})$, for all $p \in \mathbb{P}$ as follows:

$$\Phi^{\pi(w)}_w(v)(p) = \sup \left\{ \sup_{i \in \mathcal{I}} \varphi^{\pi_\lambda(w)(i,p)}_{w,i,p}(v), {X^{\text{in}}}^{\mathcal{R}}(p) \right\} \tag{12}$$

Lemma 1 (Property of $\varphi^\lambda_{i,w,p}$). *Let $\pi \in \Pi$, $w \in \text{FS}$ and $(i,p) \in \mathcal{I} \times \mathbb{P}$. The following properties are true: (I) $\varphi^{\pi_\lambda(w)(i,p)}_{w,i,p}$ is affine on $\mathbf{F}(\mathbb{P}, \mathbb{R})$; (II) $\varphi^{\pi_\lambda(w)(i,p)}_{w,i,p}$ is monotonic ; (III) $\forall\, v \in \mathbf{F}(\mathbb{P}, \mathbb{R})$, $F^{\mathcal{R}}_i(v)(p) \leq \varphi^{\pi_\lambda(w)(i,p)}_{w,i,p}(v)$; (IV) $\varphi^{\pi_\lambda(w)(i,p)}_{w,i,p}(w) = F^{\mathcal{R}}_i(w)$.*

The properties presented in Lemma 1 implies some useful properties for the maps $\Phi^{\pi(w)}_w$.

Corollary 1 (Property of $\Phi^{\pi(w)}_w$). *Let $\pi \in \Pi$ and $w \in \text{FS}$. The following properties are true: (I) $\Phi^{\pi(w)}_w$ is monotonic ; (II) $F^{\mathcal{R}} \leq \Phi^{\pi(w)}_w$; (III) $\Phi^{\pi(w)}_w(w) = F^{\mathcal{R}}(w)$; (IV) Assume that there exists $w^0 \in \text{FS}$ such that $F^{\mathcal{R}}(w^0) \leq w^0$. Then the least fixpoint of $\Phi^{\pi(w)}_w$ can be computed as the unique optimal solution of the linear program:*

$$\inf \left\{ \sum_{p' \in \mathbb{P}} v(p') \mid \forall\, (i,p) \in \mathcal{I} \times \mathbb{P},\ \varphi^{\pi_\lambda(w)(i,p)}_{i,w,p}(v) \leq v(p),\ \forall q \in \mathbb{P}, {X^{\text{in}}}^{\mathcal{R}}(q) \leq v(q) \right\}. \tag{13}$$

5.2 Policy Iteration

Now, we describe the policy iteration algorithm.

If for some k, $w^k \notin \text{FS}$ then we define $w^l = w^k$ for all $k \geq l$.

Theorem 2 (Convergence result of Algorithm 1). *The following statements hold:*

1. *For all $k \geq 0$, $F^{\mathcal{R}}(w^k) \leq w^k$;*
2. *The sequence $(w^k)_{k \geq 0}$ generated by Algorithm 1 is decreasing and converges;*
3. *Let $w^\infty = \lim_{k \mapsto +\infty} w^k$, then $F^{\mathcal{R}}(w^\infty) \leq w^\infty$. Furthermore, if $F^{\mathcal{R}}$ is upper semi-continuous and for all $k \geq \mathbb{N}$, $w^k \in \text{FS}$, then $F^{\mathcal{R}}(w^\infty) = w^\infty$.*

input : w^0, a postfixpoint of $F^{\mathcal{R}}$
output: a fixpoint $w = F^{\mathcal{R}}(w)$ if $\forall\, k \in \mathbb{N}$, $w^k \in$ FS or a postfixpoint otherwise
k=0;
while *fixpoint not reached* **do**
  begin compute the next policy π for the current iterate w^k
    if $w^k \in$ FS **then**
      Compute $F^{\mathcal{R}}(w^k)$ and define $\pi(w^k)$;
    else
      return w^k;
    end
  end
  begin compute the next iterate w^{k+1}
    Define $\Phi_{w^k}^{\pi(w^k)}$ and compute the least fixpoint w^{k+1} of $\Phi_{w^k}^{\pi(w^k)}$ from Problem (13);
    k=k+1;
  end
end

Algorithm 1: SOS-based policy iteration algorithm for PPS programs.

5.3 Initialisation and templates choice

We have supposed that we have a postfixpoint w^0 of $F^{\mathcal{R}}$. Actually this postfixpoint is *computed* at the same moment of *the templates computation*. The templates computation method can be found in [AGM15]. The method is constructed by using the definition of begin a postfixpoint of $F^{\mathcal{R}}$. Suppose that the templates basis is constituted of one template p then w^0 is a postfixpoint $F^{\mathcal{R}}$ if and only if $F^{\mathcal{R}}(w^0) \leq w^0$. This is equivalent to $\inf\{\eta \mid \eta - p + \sum_{j=1}^{n_{\text{in}}} \nu_j^{\text{in}} r_j^{\text{in}} \in \Sigma[x],\ \nu^{\text{in}} \in \Sigma[x]^{n_{\text{in}}}\} \leq w^0$ and for all $i \in \mathcal{I}$:

$$\begin{aligned}\big(F_i^{\mathcal{R}}(w^0)\big)(p) = \inf_{\lambda,\mu,\gamma,\eta} \eta & \leq w^0 \\ \text{s. t. } \begin{cases} \eta - p \circ T^i - \lambda^i(w^0 - p) + \langle \mu, r^i\rangle + \langle \gamma, r^0\rangle \in \Sigma[x] \\ \text{where } \lambda^i \geq 0,\ \mu \in \Sigma[x]^{n_i},\ \gamma \in \Sigma[x]^{n_0},\ \eta \in \mathbb{R} \end{cases}\end{aligned}$$

By definition of the infimum, it is equivalent to the existence of $\nu^{\text{in}} \in \Sigma[x]$ and for all $i \in \mathcal{I}$ of $\lambda^i \geq 0$, $\mu^i \in \Sigma[x]^{n_i}$, $\gamma^i \in \Sigma[x]^{n_0}$ such that:

$$\begin{gathered} w^0 - p + \textstyle\sum_{j=1}^{n_{\text{in}}} \nu_j^{\text{in}} r_j^{\text{in}} \in \Sigma[x] \\ w^0 - p \circ T^i - \lambda^i(w^0 - p) + \langle \mu, r^i\rangle + \langle \gamma, r^0\rangle \in \Sigma[x] \end{gathered} \tag{14}$$

Now to find a template, it suffices to solve Problem (14) where p is now a decision variable. However, two main issues appear.

First, with any objective functions, $p = 0$ is a solution of Problem (14) To avoid this very unuseful solution, we add an objective function. In [AGM15], the objective function is given by considering a proof goal i.e. we assumed that a property of the form $\mathfrak{R} \subseteq \{x \in \mathbb{R}^d \mid \kappa(x) \leq \alpha\}$ where κ is given has to be proved. Here, we are interested in proving the boundedness of the reachable value set

and we chose $\kappa = \|\cdot\|_2^2$ and we minimize α. Second, if λ^i and p are decision variables, then Problem (14) is a bilinear SOS problem which is difficult to solve and we fix $\lambda^i = 1$ as in Lyapunov equations. Note also that we can take $w^0 = 0$ since p has a constant part. In conclusion, to define a template p, we solve the following SOS program:

$$\begin{aligned}
\inf_{p\in\mathbb{R}[x]_{2m},w\in\mathbb{R}} \quad & w \ , \\
\text{s.t.} \quad & -p = \sigma_0 - \sum_{j=1}^{n_{\text{in}}} \sigma_j r_j^{\text{in}} \ , \\
& \forall\, i \in \mathcal{I},\ p - p\circ T^i = \sigma^i - \sum_{j=1}^{n_i} \mu_j^i r_j^i - \sum_{j=1}^{n_0} \gamma_j^i r_j^0 \ , \\
& w + p - \|\cdot\|_2^2 = \psi \ , \\
& \forall\, j = 1,\dots,n_{\text{in}} \ ,\ \sigma_j \in \Sigma[x] \ ,\ \deg(\sigma_j r_j^{\text{in}}) \le 2m \ , \\
& \sigma_0 \in \Sigma[x] \ ,\ \deg(\sigma_0) \le 2m \ , \\
& \forall\, i \in \mathcal{I} \ ,\ \sigma^i \in \Sigma[x] \ ,\ \deg(\sigma^i) \le 2m \deg T^i \ , \\
& \forall\, i \in \mathcal{I} \ ,\ \forall\, j = 1,\dots,n_i \ ,\ \mu_j^i \in \Sigma[x] \ ,\ \deg(\mu_j^i r_j^i) \le 2m \deg T^i \ , \\
& \forall\, i \in \mathcal{I} \ ,\ \forall\, j = 1,\dots,n_0 \ ,\ \gamma^i \in \Sigma[x] \ ,\ \deg(\gamma_j^i r_j^0) \le 2m \deg T^i \ , \\
& \psi \in \Sigma[x] \ ,\ \deg(\psi) \le 2m \ .
\end{aligned} \tag{15}$$

Let (p, w) be a solution of Problem (15). In [AGM15, Prop. 1], we proved that the set $\{x \in \mathbb{R}^d \mid p(x) \le 0\}$ defines an inductive invariant. To complete the template basis, we use the strategy proposed in [AGM15, Ex. 9], that is we work with the templates basis $\{x \mapsto x_i^2, i \in [d]\} \cup \{p\}$. We thus use the inductive invariant set $\{p(x) \le 0, x_i^2 \le w\}$ as initialisation i.e. the initial bound is $w^0(q) = w$ if $q \ne p$ and $w^(0)(q) = 0$ if $q = p$. As opposed to the approach of [AGM15], we avoid increasing the degree of polynomial p to obtain better bounds on the reachable values.

6 Experiments

Details of the running Example. Recall that our running example is given by the following PPS: $(X^{\text{in}}, X^0, \{X^1, X^2\}, \{T^1, T^2\})$, where:

$$\begin{array}{l} X^{\text{in}} = [-1,1]\times[-1,1] \\ X^0 = \mathbb{R}^d \end{array} \quad \text{and} \quad \begin{cases} X^1 = \{x \in \mathbb{R}^2 \mid -x_1^2 + 0.5 * x_2 + 0.5 \le 0\} \\ X^2 = \{x \in \mathbb{R}^2 \mid x_1^2 - 0.5 * x_2 - 0.5 < 0\} \end{cases}$$

and the functions relative to the partition $\{X^1, X^2\}$ are:

$$T^1 = \begin{pmatrix} 0.687x_1 + 0.558x_2 - 0.0001x_1x_2 \\ -0.292x_1 + 0.773x_2 \end{pmatrix} \quad T^2 = \begin{pmatrix} 0.369x_1 + 0.532x_2 - 0.0001x_1^2 \\ -1.27x_1 + 0.12x_2 - 0.0001x_1x_2 \end{pmatrix}$$

The first step consists in constructing the template basis and compute the template p and bound w on the reachable values as a solution of Problem (15). We do not give expression p for the sake of conciseness. The upper bound w

is equal to 2.1343. As suggested in Subsection 5.3, we can take the template basis $\mathbb{P} = \{p, x \mapsto x_1^2, x \mapsto x_2^2\}$. From now, we simply write $\mathbf{x}_1^2$ for $x \mapsto x_1^2$ and $\mathbf{x}_2^2$ for $x \mapsto x_2^2$. The basic semi-algebraic $\{x \in \mathbb{R}^2 \mid p(x) \leq 0,\ x_1^2 \leq 2.1343,\ x_2^2 \leq 2.1343\}$ is an inductive invariant and the corresponding bounds function is $w^0 = (w^0(\mathbf{x}_1^2), w^0(\mathbf{x}_2^2), w^0(p)) = (0, 2.1343, 2.1343)$.

As in first step of Algorithm 1, we compute the image of w^0 by $F^{\mathcal{R}}$. We found that $F^{\mathcal{R}}(w^0)(x \mapsto x_1^2) = 1.5503,\ F^{\mathcal{R}}(w^0)(x \mapsto x_2^2) = 1.9501$ and $F^{\mathcal{R}}(w^0)(p) = 0$. The computation of $F^{\mathcal{R}}$ permits to determine a new policy. The important data is the vector λ. For example, for $i = 1$ and the template $\mathbf{x}_1^2$, the vector λ is $(0, 0, 2.0332)$. It means that we associate for each template q a weight $\lambda(q)$. In the case of $\lambda = (0, 0, 2.0332)$, $\lambda(\mathbf{x}_1^2) = 0$, $\lambda(\mathbf{x}_2^2) = 0$ and $\lambda(p) = 2.0332$. For $i = 1$, the template $\mathbf{x}_1^2$ and the bound vector w^0, the function $\varphi^{\lambda}_{w^0,1,\mathbf{x}_1^2}(v) = 2.0332v(p) + 1.5503$.

To get the new invariant, we compute a bound vector w^1 solution of Linear Program (13). We obtain: $w^1(\mathbf{x}_1^2) = 1.5503,\ w^1(\mathbf{x}_2^2) = 1.9501$ and $w^1(p) = 0$. Since we have $\|F^{\mathcal{R}}(w^1) - w^1\|_\infty \leq 1e-6$, Policy iteration terminates with this fixpoint.

Benchmarks. The presented analysis has been applied to available examples of the control community literature: piecewise linear systems, polynomial systems, etc. We gathered the examples matching our criteria: discrete systems, possibly piecewise, at most polynomial. In all the considered cases, no common quadratic Lyapunov existed. In other words, not only the existing linear abstractions such as intervals or polyhedra would fail in computing a non trivial postfixpoint, but also the existing analyses dedicated to digital filters such as [Fer04,GS07,AGG11,RJGF12].

The analysis has been implemented in Matlab and relies on the Mosek SDP solver [AA00], through the Yalmip [L̈04] SOS front-end. Without outstanding performances, all experiments are performed within a few seconds per iteration, which makes us believe that a more serious implementation would perform better. We recall that the analysis could be interrupted at any point, still providing a safe upper bound.

The table 1 in appendix summarizes the examples considered, the bounds obtained, the degree of the polynomial templates used and the number of iterations performed before reaching the fixpoint.

7 Conclusion

In this paper, we extend the previous policy iteration algorithm semidefinite programming based to a sum-of-squares programming setting. This extention allows to consider the wider class of programs written in polynomial arithmetics and composed of a single loop with a nested conditional branchments loop body. We have proved that, in this new setting, we keep the advantages of policy iteration algorithms, that is, they produce a sequence of more and more precise safe overapproximations of the reachable values set.

As future works, we could generalize our technique to programs manipulating semialgebraic or transendantal arithmetics.

References

[AA00] Erling D. Andersen and Knud D. Andersen. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High Performance Optimization*, volume 33 of *Applied Optimization*, pages 197–232. Springer, 2000.

[AGG10] A. Adjé, S. Gaubert, and E. Goubault. Coupling policy iteration with semidefinite relaxation to compute accurate numerical invariants in static analysis. In *ESOP*, volume 6012 of *LNCS*, pages 23–42. Springer, 2010.

[AGG11] A. Adjé, S. Gaubert, and E. Goubault. Coupling policy iteration with semidefinite relaxation to compute accurate numerical invariants in static analysis. *Logical Methods in Computer Science*, 8(1), 2011.

[AGM15] A. Adjé, P.-L. Garoche, and V. Magron. Property-based Polynomial Invariant Generation using Sums-of-Squares Optimization, 2015. submitted.

[AJ13] Amir Ali Ahmadi and Raphael M. Jungers. Switched stability of nonlinear systems via sos-convex lyapunov functions and semidefinite programming. In *CDC 2013*, pages 727–732, 2013.

[CC77] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252. ACM Press, New York, NY, 1977.

[CGG+05] A. Costan, S. Gaubert, E. Goubault, M. Martel, and S. Putot. A policy iteration algorithm for computing fixed points in static analysis of programs. In *Computer aided verification*, pages 462–475. Springer, 2005.

[DP02] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, New York, second edition, 2002.

[Fen02] Gang Feng. Stability analysis of piecewise discrete-time linear systems. *IEEE Trans. Automat. Contr.*, 47(7):1108–1112, 2002.

[Fer04] Jérôme Feret. Static analysis of digital filters. In *ESOP 2004*, volume 2986 of *LNCS*, pages 33–48. Springer, 2004.

[GGTZ07] Stephane Gaubert, Eric Goubault, Ankur Taly, and Sarah Zennou. Static analysis by policy iteration on relational domains. In *ESOP 2007*, volume 4421 of *LNCS*, pages 237–252. Springer, 2007.

[GS07] Thomas Gawlitza and Helmut Seidl. Precise fixpoint computation through strategy iteration. In *ESOP 2007*, volume 4421 of *LNCS*, pages 300–315. Springer, 2007.

[GSA+12] Thomas Martin Gawlitza, Helmut Seidl, Assalé Adjé, Stéphane Gaubert, and Eric Goubault. Abstract interpretation meets convex optimization. *J. Symb. Comput.*, 47(12):1416–1446, 2012.

[LÖ4] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[Las09] Jean-Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

[Mor70] J. J. Moreau. Inf-convolution, sous-additivité, convexité des fonctions numériques. *Journal de Mathématiques Pures et Appliquées*, 49:109–154, 1970.

[RJGF12] P. Roux, R. Jobredeaux, P-L. Garoche, and E. Feron. A generic ellipsoid abstract domain for linear time invariant systems. In T. Dang and I. M. Mitchell, editors, *HSCC*, pages 105–114. ACM, 2012.

[Roc96] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

[Rub00] A. M. Rubinov. *Abstract Convexity and Global optimization*. Kluwer Academic Publishers, 2000.

[Sin97] I. Singer. *Abstract Convex Analysis*. Wiley-Interscience Publication, 1997.

[SSM04] Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Constraint-based linear-relations analysis. In *SAS 2004*, volume 3148 of *LNCS*, pages 53–68. Springer, 2004.

A Appendix

A.1 Benchmarks

Reference	Bounds (ie. x_i^2)	Degree	# it.
[Fen02, Ex. 2.1] piecewise linear with 2 zones	$[3.8260, 2.1632, 1.0000]$	4	1
	$[3.7482, 1.8503, 1.0000]$	6	1
	$[1.0000, 1.8709, 1.0000]$	8	max (10)
	No good invariant	10,12	–
[Fen02, Ex. 3.3] piecewise linear with 4 zones	$[1.8359, 1.3341]$	4	2
	$[1.5854, 1.2574]$	6	5
	$[1.5106, 1.2569]$	8	4
	$[1.4813, 1.2544]$	10	6
[AJ13, Ex. 3] piecewise quadratic with 2 zones	No good invariant	4	–
	$[1.5503, 1.9501]$	6	1
	$[1.5503, 1.9502]$	8	7
	$[1.5500, 1.9436]$	10	1
	$[1.5503, 1.9383]$	12	2
Hand-crafted piecewise polynomial (deg 3) with 2 zones	No good invariant	4,6,8,10	–
	$[1.2100, 0.9989]$	12	max (10)

"No good invariant" occurs when the template synthesis fails, ie. does not provide a sound postfixpoint. It seems to be caused both by the large size of the SDP problems generated and by numerical inconsistencies of the interior point method used in the solvers.

Table 1. Experiments

[Fen02, Ex. 2.1] piecewise linear with 2 zones

$$X^{\text{in}} = [-1, 1]^3$$
$$X^0 = \mathbb{R}^3$$
$$X^1 = \{(x, y, z) \in \mathbb{R}^3 | x \leq 0\} \quad X^2 = \{(x, y, z) \in \mathbb{R}^3 | x > 0\}$$
$$T^1 = \begin{pmatrix} x + 0.5y \\ -0.3x + 0.8y \\ 0.4z \end{pmatrix} \qquad T^2 = \begin{pmatrix} x + .4y + 0.01z \\ -0.1x + 0.8y \\ 0.5z \end{pmatrix}$$

[Fen02, Ex. 3.3] piecewise linear with 4 zones

$$
\begin{aligned}
&X^{\text{in}} = [-1,1]^2\\
&X^0 = \mathbb{R}^2\\
&X^1 = \{(x,y) \in \mathbb{R}^2 | x \leq -1\} && X^2 = \{(x,y) \in \mathbb{R}^2 | x \in]-1,1] \wedge y > 0\}\\
&X^3 = \{(x,y) \in \mathbb{R}^2 | x \in]-1,1] \wedge y \leq 0\} && X^4 = \{(x,y) \in \mathbb{R}^2 | x > 1\}\\
&T^1 = \begin{pmatrix} 0.9x - 0.01y \\ 0.1x + y - 0.02 \end{pmatrix} && T^4 = \begin{pmatrix} 0.9x - 0.01y \\ 0.1x + y + 0.02 \end{pmatrix}\\
&T^2 = T^3 = \begin{pmatrix} x - 0.02y \\ 0.02x + 0.9y \end{pmatrix}
\end{aligned}
$$

[AJ13, Ex. 3] piecewise quadratic with 2 zones

$$
\begin{aligned}
&X^{\text{in}} = [-1,1]^2\\
&X^0 = \mathbb{R}^2\\
&X^1 = \{(x,y) \in \mathbb{R}^2 | 1 \leq x^2\} && X^2 = \{(x,y) \in \mathbb{R}^2 | x^2 < 1\}\\
&T^1 = \begin{pmatrix} 0.687x + 0.558y - 0.0001xy \\ -0.292x + 0.773y \end{pmatrix} && T^2 = \begin{pmatrix} 0.369x + 0.532y - 0.0001x^2 \\ -1.27x + 0.12y - 0.0001xy \end{pmatrix}
\end{aligned}
$$

Hand-crafted piecewise polynomial (deg 3) with 2 zones

$$
\begin{aligned}
&X^{\text{in}} = [0.9, 1.1] \times [0, 0.2]\\
&X^0 = \mathbb{R}^2\\
&X^1 = \{(x,y) \in \mathbb{R}^2 | x^2 + y^2 \leq 1\} && X^2 = \{(x,y) \in \mathbb{R}^2 | x^2 + y^2 > 1\}\\
&T^1 = \begin{pmatrix} x^2 + y^3 \\ x^3 + y^2 \end{pmatrix} && T^2 = \begin{pmatrix} 0.5x^3 + 0.4y^2 \\ -0.6x^2 + 0.3y^2 \end{pmatrix}
\end{aligned}
$$

A.2 Proofs

Proof (Proof of Lemma 1). Let $w \in \text{FS}$, $i \in \mathcal{I}$, $p \in \mathbb{P}$ and $\pi \in \Pi$.

(I) The fact that $\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}$ is affine follows readily from the definition.

(II) The monotonicity of $\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}$ follows from the positivity of $\pi_\lambda(w)(i,p)$.

(III) Let $v \in \mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$. Since $w \in \text{FS}$, there exists $(\lambda, \mu, \gamma, g) \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right) \times \Sigma[x]^{n_i} \times \Sigma[x]^{n_0} \times \mathbb{R}[x]^l$ such that

$$\left(F_i^{\mathcal{R}}(w)\right)(p) = p \circ T^i + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q) - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \sum_{j=1}^{l} g_j^2$$

Thus:

$$
\begin{aligned}
\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}(v) &= \textstyle\sum_{q \in \mathbb{P}} \lambda(q) v(q) - \sum_{q \in \mathbb{P}} \lambda(q) w(q) + p \circ T^i + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q)\\
&\quad - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \textstyle\sum_{j=1}^{l} g_j^2\\
&= p \circ T^i + \textstyle\sum_{q \in \mathbb{P}} \lambda(q)(v(q) - q) - \langle \mu, r^i \rangle - \langle \gamma, r^0 \rangle + \sum_{j=1}^{l} g_j^2
\end{aligned}
$$

And finally,

$$\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}(v) - p \circ T^i - \sum_{q\in\mathbb{P}} \lambda(q)(v(q)-q) + \langle \mu, r^i\rangle + \langle \gamma, r^0\rangle \in \Sigma[x] \qquad (16)$$

and recall that :

$$\big(F_i^{\mathcal{R}}(v)\big)(p) = \inf_{\lambda,\mu,\gamma,\eta} \eta$$
$$\text{s. t.} \begin{cases} \eta - p\circ T^i - \sum_{q\in\mathbb{P}} \lambda(q)(v(q)-q) + \langle \mu, r^i\rangle + \langle \gamma, r^0\rangle \in \Sigma[x] \\ \text{where } \lambda \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}_+\right),\ \mu \in \Sigma[x]^{n_i},\ \gamma \in \Sigma[x]^{n_0},\ \eta \in \mathbb{R} \end{cases}$$

And from Equation (16), $(\lambda, \mu, \gamma, \varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}(v))$ is a feasible solution of the latter minimization problem and we conclude that $\big(F_i^{\mathcal{R}}(v)\big)(p) \leq \varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}(v)$.

(IV)

$$\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}(w) = \sum_{q\in\mathbb{P}} \lambda(q)w(q) + \big(F_i^{\mathcal{R}}(w)\big)(p) - \sum_{q\in\mathbb{P}} \lambda(q)w(q) = \big(F_i^{\mathcal{R}}(w)\big)(p)\ .$$

Proof (Proof of Corollary 1). Let $\pi \in \Pi$ and $w \in \mathrm{FS}$.

(I) $\Phi_w^{\pi(w)}$ is monotonic from the monotonicity of the map $\varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}$ for all $i \in \mathcal{I}$ and for all $p \in \mathbb{P}$, and the the fact that the pointwise supremum of monotonic maps is also monotonic.

(II) Let $v \in \mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$ and let $p \in \mathbb{P}$. Recall that:

$$\big(F^{\mathcal{R}}(v)\big)(p) = \sup\left\{\sup_{i\in\mathcal{I}} \big(F_i^{\mathcal{R}}(v)\big)(p), X^{\mathrm{in}\,\mathcal{R}}(p)\right\}$$

and from the second assertion of Lemma 1, we have for all $i \in \mathcal{I}$, $F_i^{\mathcal{R}}(v)(p) \leq \varphi_{w,i,p}^{\pi_\lambda(w)(i,p)}$, by taking the supremum over $\mathcal{I}$ and then the supremum with $X^{\mathrm{in}\,\mathcal{R}}(p)$, we obtain that $F^{\mathcal{R}}(v)(p) \leq \Phi_w^{\pi(w)}(v)(p)$ that is the desired result.

(III) This result follows readily from the third assertion of Lemma 1.

(IV) By Tarski's theorem and from the monotonicity of $\Phi_w^{\pi(w)}$, $\Phi_w^{\pi(w)}$ has a least fixpoint in $\mathbf{F}\left(\mathbb{P}, \overline{\mathbb{R}}\right)$. Let L be this least fixpoint. Let $w^0 \in \mathrm{FS}$ such that $F^{\mathcal{R}}(w^0) \leq w^0$. This implies that $\Phi_w^{\pi(w)}(w^0) = F^{\mathcal{R}}(w^0) \leq w^0$ and thus L cannot take the value $+\infty$. Moreover from the definition of $\Phi_w^{\pi(w)}$, $L \geq X^{\mathrm{in}\,\mathcal{R}}$ which is finite and thus $L \in \mathbf{F}\left(\mathbb{P}, \mathbb{R}\right)$. Now, from Tarski's theorem and the definition of $\Phi_w^{\pi(w)}$, we have:

$$\begin{aligned} L &= \min\{v \mid \Phi_w^{\pi(w)}(v) \leq v\} \\ &= \inf\left\{v \mid \forall (i,p) \in \mathcal{I}\times\mathbb{P},\ \varphi_{i,w,p}^{\pi_\lambda(w)(i,p)}(v) \leq v(p),\ \forall q \in \mathbb{P}, X^{\mathrm{in}\,\mathcal{R}}(q) \leq v(q)\right\}\ . \end{aligned}$$

Let us suppose that there exists a feasible solution $\bar{v}$ such that $\sum_{q'\in\mathbb{P}} \bar{v}(q') < \sum_{q'\in\mathbb{P}} L(q')$. Then we have $\inf\{\bar{v}, L\} \leq L$ and $\inf\{\bar{v}, L\} \neq L$. From the monotonicity of $\Phi_w^{\pi(w)}$ and the feasibility of $\bar{v}$ and L, we have $\Phi_w^{\pi(w)}(\inf\{\bar{v}, L\}) \leq \inf\{\bar{v}, L\}$. This contradicts the minimality of L. We conclude that L is the optimal solution of Linear Program (13).

Proof (Proof of Proposition 3). Assuming $p, \lambda, \mu, \gamma, \eta$ fixed, we denote by $\psi(w)$ the polynomial in x, $\eta - p(T^i(x)) - \sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)(w(q) - q(x)) + \langle \mu, r^i\rangle(x) + \langle \gamma, r^0\rangle(x)$. Assume that $w \leq_{\mathbf{F}} w'$. We have $\forall q \in \mathbb{P}, w(q) = w(q) - w'(q) + w'(q)$ and thus $-\sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)w(q) = -\sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)w'(q) - \sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)(w(q) - w'(q))$. Now we remark that $\psi(w') = \psi(w) + \sum_{\mathrm{q}\in\mathbb{P}} \lambda(q)(w(q) - w'(q))$. Then if $\psi(w)$ is a SOS, so does $\psi(w')$. Finally, we recall that if $A \subseteq B$, then $\inf_B \leq \inf_A$. We conclude that $\big(F_i^{\mathcal{R}}(w)\big)(p) \leq \big(F_i^{\mathcal{R}}(w')\big)(p)$.

Proof (Proof of Theorem 2). We prove the first point by induction. We have $F^{\mathcal{R}}(w^0) \leq w^0$ by assumption. Now suppose that for some $k \in \mathbb{N}$, $F^{\mathcal{R}}(w^k) \leq w^k$. If $w^k \notin \mathrm{FS}$ then $w^l = w^k$ for all $l \geq k$ and then we have proved the result. Now suppose that $w^k \in \mathrm{FS}$ and let us take $\pi \in \mathit{\Pi}$. From the second point of Corollary 1, $F^{\mathcal{R}}(w^{k+1}) \leq \mathit{\Phi}_{w^k}^{\pi(w^k)}(w^{k+1})$ and since w^{k+1} is the least fixpoint of $\mathit{\Phi}_{w^k}^{\pi(w^k)}(w^{k+1})$ then $F^{\mathcal{R}}(w^{k+1}) \leq w^{k+1}$.

Let us prove the second assertion. Let $k \geq 0$. If $w^k \notin \mathrm{FS}$ then $w^{k+1} = w^k \leq w^k$. Now suppose that $w^k \in \mathrm{FS}$ and let $\pi \in \mathit{\Pi}$, then from the last point of Corollary 1, $\mathit{\Phi}_{w^k}^{\pi(w^k)}(w^k) = F^{\mathcal{R}}(w^k) \leq w^k$ from the first point. Then w^k is a feasible solution of Problem (13) and since w^{k+1} is the optimal solution of Problem (13) then $w^{k+1} \leq w^k$. We have $X^{\mathrm{in}^{\mathcal{R}}} \leq F^{\mathcal{R}}(w^k)$ for all $k \in \mathbb{N}$, then $(w^k)_{k\geq 0}$ is decreasing and lower bounded then it converges to some w^∞.

Let us prove the last assertion. If for some k, $w^k \notin \mathrm{FS}$, then $w^\infty = w^k$ and we have $F^{\mathcal{R}}(w^\infty) \leq w^\infty$ from the first point. Now suppose that for all $k \in \mathbb{N}$, $w^k \in \mathrm{FS}$. Since $F^{\mathcal{R}}$ is monotonic then for all $k \in \mathbb{N}$, $F^{\mathcal{R}}(w^\infty) \leq F^{\mathcal{R}}(w^k) \leq w^k$ from the first point. Now taking the limit of the right-hand side, we get $F(w^\infty) \leq w^\infty$. Now, let $k \in \mathbb{N}$ and let $\pi \in \mathit{\Pi}$. From the second point, $w^{k+1} \leq w^k$ and from the monotonicity of $\mathit{\Phi}_{w^k}^{\pi(w^k)}$, we have $w^{k+1} = \mathit{\Phi}_{w^k}^{\pi(w^k)}(w^{k+1}) \leq \mathit{\Phi}_{w^k}^{\pi(w^k)}(w^k) = F^{\mathcal{R}}(w^k)$. By taking the $\limsup$ on k, we get $w^\infty \leq \limsup_{k\to+\infty} F^{\mathcal{R}}(w^k)$. If $F^{\mathcal{R}}$ is upper semicontinuous then $w^\infty \leq \limsup_{k\to+\infty} F^{\mathcal{R}}(w^k) \leq F^{\mathcal{R}}(w^\infty)$ and $w^\infty = F^{\mathcal{R}}(w^\infty)$.