Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis

Assalé Adjé[+], Stéphane Gaubert[*] and Eric Goubault[+]

[+] CEA LIST MeASI Saclay and LIX, Ecole Polytechnique
[*] INRIA Saclay and CMAP, Ecole Polytechnique

19th European Symposium Of Programming
March 22-26, 2010, Paphos, Cyprus

Computing numerical invariant is an important problem.

A lot of related works (non exhaustive list) in static analysis of numerical programs:

Bagnara	SAS 20005
Cousot et al	Astrée
Feret	NSAD 2005
Feron	arxiv:012.1986 2008
Goubault/Putot	SAS 2006, CAV 2009, CAV 2010
Manna/Sankaranarayanan	VMCAI 2005
Miné	VMCAI 2005
Monniaux	CAV 2005

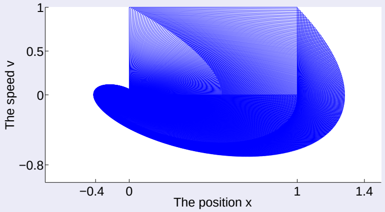
Technique used: Galois based abstract interpretation to compute overapproximation of invariants.

Running example

We consider an harmonic oscillator $\ddot{x} + c\dot{x} + x = 0$.

Figure: An harmonic oscillator, its Euler integration scheme and the loop invariant found at control point 2

```
x = [0,1];
v = [0,1];
h = 0.01;
c = 1;  [1]
while (true) {  [2]
  ov = v;
  v = v*(1-hc)-h*x;
  x = x+h*ov;  [3] }
```



Trajectories of the Euler scheme integration

Running example

We consider an harmonic oscillator $\ddot{x} + c\dot{x} + x = 0$.

Figure: An harmonic oscillator, its Euler integration scheme and the loop invariant found at control point 2

```
x = [0,1];
v = [0,1];
h = 0.01;
c = 1;  [1]
while (true) {  [2]
   ov = v;
   v = v*(1-hc)-h*x;
   x = x+h*ov;  [3] }
```

Convex polyhedra fail:

$x = \top, \; v = \top$

$h = 0.01, \; c = 1$

(e.g Apron-interproc)

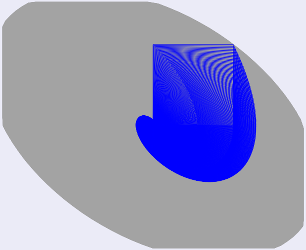
Running example

We consider an harmonic oscillator $\ddot{x} + c\dot{x} + x = 0$.

Figure: An harmonic oscillator, its Euler integration scheme and the loop invariant found at control point 2

```
x = [0,1];
v = [0,1];
h = 0.01;
c = 1;  [1]
while (true) {  [2]
   ov = v;
   v = v*(1-hc)-h*x;
   x = x+h*ov;  [3]  }
```

This talk:



$$\left\{ \begin{array}{c} -1.8708 \leq x \leq 1.8708 \\ -1.5275 \leq v \leq 1.5275 \\ 2x^2 + 3v^2 + 2xv \leq 7 \end{array} \right\}$$

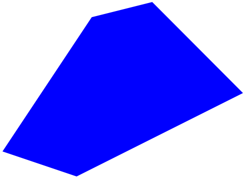
First ingredient:
Non linear templates

Linear templates

Domain of polyhedra (Cousot,Halbwachs 78):

Problem: number of extreme points and faces blows up!
An approach: Manna, Sankaranarayanan, and Sipma (VMCAI05).
Polyhedra with prescribed normals of facets so no exponential blow up.

The user provided linear templates: $p_1(x,y) = \frac{1}{2}x - y$, $p_2(x,y) = y - \frac{3}{2}x$, $p_3(x,y) = y - \frac{1}{4}x$, $p_4(x,y) = y + x$ and $p_5(x,y) = -y - \frac{1}{3}x$



$$\left\{ (x,y) \;\middle|\; \begin{array}{lcl} p_1(x,y) & \leq & 1 \\ p_2(x,y) & \leq & 5 \\ p_3(x,y) & \leq & \frac{5}{2} \\ p_4(x,y) & \leq & 2 \\ p_5(x,y) & \leq & 3 \end{array} \right\}$$

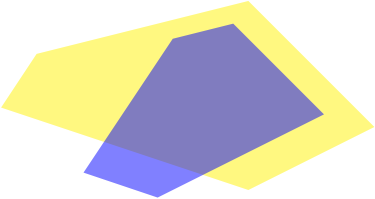
Figure: On the left, the geometric concretization of the abstract set on the right

Linear templates

Domain of polyhedra (Cousot,Halbwachs 78):

Problem: number of extreme points and faces blows up!
An approach: Manna, Sankaranarayanan, and Sipma (VMCAI05).
Polyhedra with prescribed normals of facets so no exponential blow up.

The user provided linear templates: $p_1(x,y) = \frac{1}{2}x - y$, $p_2(x,y) = y - \frac{3}{2}x$, $p_3(x,y) = y - \frac{1}{4}x$, $p_4(x,y) = y + x$ and $p_5(x,y) = -y - \frac{1}{3}x$



$$\left\{ (x,y) \;\middle|\; \begin{array}{rcr} p_1(x,y) & \leq & 2 \\ p_2(x,y) & \leq & 10 \\ p_3(x,y) & \leq & 3 \\ p_4(x,y) & \leq & 3 \\ p_5(x,y) & \leq & 2 \end{array} \right\}$$

Figure: On the left, the geometric concretization of the abstract set on the right

Linear templates

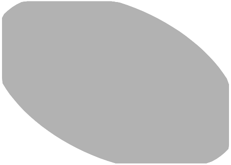
Domain of polyhedra (Cousot,Halbwachs 78):

Problem: number of extreme points and faces blows up!
An approach: Manna, Sankaranarayanan, and Sipma (VMCAI05).
Polyhedra with prescribed normals of facets so no exponential blow up.

Special cases:
Linear templates are $\pm e_i$, we get intervals.
Linear templates are $\pm(e_i - e_j)$, we get zones.

Non linear templates

Taking $p_1(x,v) = x$, $p_2(x,v) = v$ and $p_3(x,v) = 2x^2 + 3v^2 + 2xv$.



$$\left\{ (x,y) \mid \begin{array}{rcr} p_1(x,y) & \leq & 1.8708 \\ -p_1(x,y) & \leq & 1.8708 \\ p_2(x,y) & \leq & 1.5275 \\ -p_2(x,y) & \leq & 1.5275 \\ p_3(x,y) & \leq & 7 \end{array} \right\}$$

Figure: On the left, the geometric concretization of the abstract set on the right

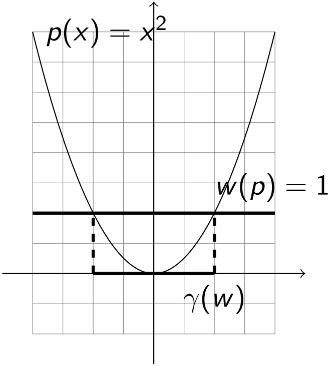
In all these sets, we fix a set of functions $P = \{p_1, p_2, \ldots, p_m\}$ and we give "bounds" on these functions.

Concretization map

P set of functions from $\mathbb{R}^d$ to $\mathbb{R}$ and $\mathcal{F}(P, \overline{\mathbb{R}})$ set of functions from P to $\overline{\mathbb{R}}$.

For $w \in \mathcal{F}(P, \overline{\mathbb{R}})$, we define the concretization map:

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

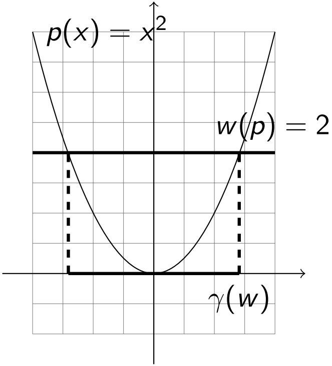


Concretization map

P set of functions from $\mathbb{R}^d$ to $\mathbb{R}$ and $\mathcal{F}(P,\overline{\mathbb{R}})$ set of functions from P to $\overline{\mathbb{R}}$.

For $w \in \mathcal{F}(P,\overline{\mathbb{R}})$, we define the concretization map:

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

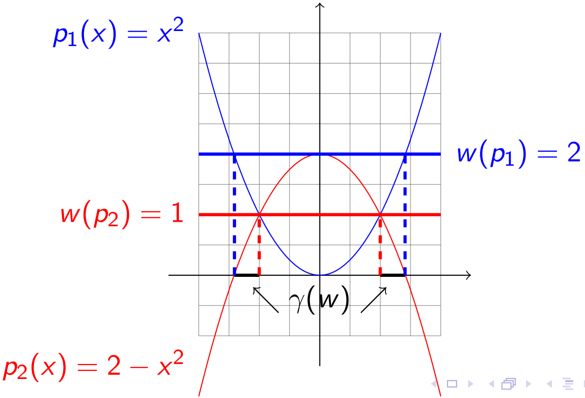


Concretization map

P set of functions from $\mathbb{R}^d$ to $\mathbb{R}$ and $\mathcal{F}(P, \overline{\mathbb{R}})$ set of functions from P to $\overline{\mathbb{R}}$.

For $w \in \mathcal{F}(P, \overline{\mathbb{R}})$, we define the concretization map:

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



Concretization map

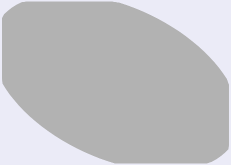
P set of functions from $\mathbb{R}^d$ to $\mathbb{R}$ and $\mathcal{F}(P,\overline{\mathbb{R}})$ set of functions from P to $\overline{\mathbb{R}}$.

For $w \in \mathcal{F}(P,\overline{\mathbb{R}})$, we define the concretization map:

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

Example

$P = \{p_1, -p_1, p_2, -p_2, p_3\}$ with $p_1(x,v) = x$, $p_2(x,v) = v$ and $p_3(x,v) = 2x^2 + 3v^2 + 2xv$.
$w(p_1) = w(-p_1) = 1.8708$, $w(p_2) = w(-p_2) = 1.5275$ and $w(p_3) = 7$.

$\gamma(w) =$ 

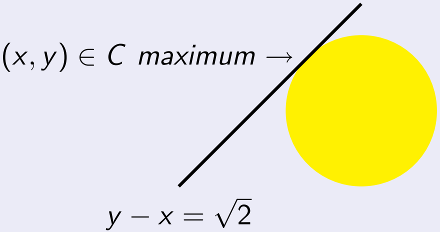
Abstraction map

For $C \subset \mathbb{R}^d$, we define the abstraction map:

$$\boxed{\begin{array}{rccl} \alpha(C): & P & \rightarrow & \overline{\mathbb{R}} \\ & p & \mapsto & \sup\{p(x) \mid x \in C\} \end{array}}$$

Example

$C = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ *and*
$P = \{p_1(x, y) = x, p_2(x, y) = y, p_3(x, y) = y - x\}$.



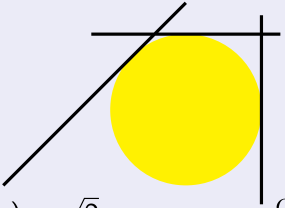
Abstraction map

For $C \subset \mathbb{R}^d$, we define the abstraction map:

$$\boxed{\begin{array}{llll} \alpha(C): & P & \to & \overline{\mathbb{R}} \\ & p & \mapsto & \sup\{p(x) \mid x \in C\} \end{array}}$$

Example

$C = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ *and*
$P = \{p_1(x, y) = x, p_2(x, y) = y, p_3(x, y) = y - x\}$.



A Galois connection

We recall that,
for $w \in \mathcal{F}(P, \overline{\mathbb{R}})$,

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

and for $C \subset \mathbb{R}^d$,

$$\begin{array}{rrcl} \alpha(C): & P & \rightarrow & \overline{\mathbb{R}} \\ & p & \mapsto & \sup\{p(x) \mid x \in C\} \end{array}$$

We get:

Proposition

(α, γ) *defines a Galois connection between* $\mathcal{F}(P, \overline{\mathbb{R}})$ *and* $\mathcal{P}(\mathbb{R}^d)$.

Second ingredient: Semidefinite positive relaxation

Quadratic templates

P is a finite set of functions:

Quadratic templates

$$p \mapsto x^T A_p x + x^T b_p + c_p$$

where A_p are $d \times d$ symmetric matrices, b_p are $\mathbb{R}^d$ vectors and c_p are scalars.

For instance:

$P = \{p_1, -p_1, p_2, -p_2, p_3\}$ with $p_1(x, v) = x$, $p_2(x, v) = v$ and $p_3(x, v) = 2x^2 + 3v^2 + 2xv$ is a set of quadratic templates.

Compute bounds in abstract interpretation

In abstract interpretation, we solve classically: $F^\sharp(w) = \alpha \circ F \circ \gamma(w) = w$. So,

$$(F^\sharp(w))(p) = \sup\{p(y) \mid y = F(x),\ x \in \gamma(w)\}$$

For a linear assignment $y := Mx$, assume that $q(x) \leq w(q),\ \forall q \in P$. In quadratic templates, we have:

$$\boxed{\begin{array}{rll} (F^\sharp(w))(p) = & \sup & x^T M^T A_p M x + x^T M^T b_p + c_p \\ & \text{s.t} & x^T A_q x + x^T b_q + c_q \leq w(q) \\ & & \forall q \in P \end{array}}$$

Can be solved in polynomial time when every templates p are both concave and convex i.e linear.

Compute bounds in abstract interpretation

In abstract interpretation, we solve classically: $F^\sharp(w) = \alpha \circ F \circ \gamma(w) = w$. So,

$$(F^\sharp(w))(p) = \sup\{p(y) \mid y = F(x),\ x \in \gamma(w)\}$$

For a linear assignment $y := Mx$, assume that $q(x) \leq w(q)$, $\forall q \in P$. In quadratic templates, we have:

$$\boxed{\begin{array}{lll} (F^\sharp(w))(p) = & \sup & x^T M^T A_p M x + x^T M^T b_p + c_p \\ & \text{s.t} & x^T A_q x + x^T b_q + c_q \leq w(q) \\ & & \forall q \in P \end{array}}$$

In general this problem is NP-Hard [PV91]!

Compute bounds in abstract interpretation

In abstract interpretation, we solve classically: $F^\sharp(w) = \alpha \circ F \circ \gamma(w) = w$. So,

$$(F^\sharp(w))(p) = \sup\{p(y) \mid y = F(x),\ x \in \gamma(w)\}$$

For a linear assignment $y := Mx$, assume that $q(x) \leq w(q)$, $\forall q \in P$. In quadratic templates, we have:

$$\boxed{\begin{array}{rll} (F^\sharp(w))(p) = & \sup & x^T M^T A_p M x + x^T M^T b_p + c_p \\ & \text{s. t} & x^T A_q x + x^T b_q + c_q \leq w(q) \\ & & \forall q \in P \end{array}}$$

However we propose to compute an overapproximation in polynomial time (under technical hypothesis very often satisfied!) by the Shor relaxation scheme.

Concepts of Lagrange Duality (1)

$g, g_1, \ldots, g_n$ be functions from $\mathbb{R}^d$ to $\mathbb{R}$.
A constrained optimization problem:

$$\begin{array}{cc} \sup & g(x) \\ \mathrm{s.t} & g_1(x) \leq 0 \\ & \vdots \\ & g_n(x) \leq 0 \end{array}$$

We write: $L(\lambda, x) = g(x) - \sum_{i=1}^{n} \lambda_i g_i(x)$ and we remark that:

$$\inf_{\lambda \in \mathbb{R}^n_+} L(\lambda, x) = \begin{cases} g(x) & \text{if } \forall\, i \; g_i(x) \leq 0 \\ -\infty & \text{otherwise} \end{cases}$$

The entries of vectors $\lambda \in \mathbb{R}^n_+$ are called Lagrange multipliers.

Concepts of Lagrange Duality (2)

g concave, $g_1, \ldots, g_n$ convex and if there exists x s.t $g_i(x) < 0$ (Strong duality theorem):

$$\inf_{\lambda \in \mathbb{R}^n_+} \sup_{x \in \mathbb{R}^d} L(x, \lambda) = \sup\{g(x) \mid g_i(x) \leq 0 \ \forall i\}$$

In quadratic templates, to apply this theorem, the templates p should be linear!

However:

In general, we just have an overapproximation (Weak duality theorem):

$$\boxed{\inf_{\lambda \in \mathbb{R}^n_+} \sup_{x \in \mathbb{R}^d} \left(g(x) - \sum_{i=1}^{n} \lambda_i g_i(x) \right) \geq \sup\{g(x) \mid g_i(x) \leq 0 \ \forall i\}}$$

In our case: $F^{\mathcal{R}} = \inf F^{\lambda}$ with $F^{\lambda} = \sup_{x \in \mathbb{R}^d} \left(g(x) - \sum_{i=1}^{n} \lambda_i g_i(x) \right)$.

Shor relaxation

For quadratic templates: $(F^{\mathcal{R}}(w))(p)$ can be rewritten as [Sho87, TN01]:

Minimize	a linear function
s. t	- a linear combination of matrices is SDP - the coefficient of combination are Lagrange multipliers

($SDP =$ Semi-definite positive i.e all eigenvalues are nonnegative.)

This minimization problem can be solved in polynomial time [NN94].

Shor relaxation

For quadratic templates: $(F^{\mathcal{R}}(w))(p)$ can be rewritten as [Sho87, TN01]:

$$\begin{array}{ll} \text{Minimize} & \eta \\ \text{s.t} & N(p \circ M) + \eta R(-1) + \sum_{q \in P} \lambda(q)[R(w(q)) - N(q)] \text{ is SDP} \\ & \lambda(q) \geq 0,\ \forall\, q \in P \\ & \eta \in \mathbb{R} \end{array}$$

Where $N(p) = \begin{pmatrix} c_p & \frac{1}{2} b_p^T \\ \frac{1}{2} b_p & A_p \end{pmatrix}$, and for $y \in \mathbb{R}$, $R_{1,1}(y) = y$ and 0 otherwise.

This minimization problem can be solved in polynomial time [NN94].

Third ingredient: Policy iteration

Fixpoint computation

Our semantic, for:

- Linear assignment and intersection(tests): we compute an overapproximation of abstract semantic $F^\sharp$ using Shor relaxation.
- Union: interpreted as a sup (no relaxation).

Fixpoint computation

We write $\underline{x}(x,v) = x$, $\underline{v}(x,v) = v$, $\underline{E}(x,v) = 2x^2 + 3v^2 + 2xv$ and $P = \{\underline{x}, -\underline{x}, \underline{v}, -\underline{v}, \underline{E}\}$.
For harmonic oscillator, the abstract semantic $F^\sharp$, given by Galois connection:

$$\boxed{\begin{array}{rcl} F_1^\sharp(w)(p) & = & \{0 \leq \underline{x}(x,v) \leq 1,\, 0 \leq \underline{v}(x,v) \leq 1,\, \underline{E}(x,v) \leq 7\} \\ F_2^\sharp(w)(p) & = & \sup\{w_1(p), w_3(p)\} \\ F_3^\sharp(w)(p) & = & \sup\limits_{(x,v)\in\gamma(w_2)} p(M(x,v)) \end{array}}$$

with $\begin{pmatrix} 1 & h \\ -h & 1-hc \end{pmatrix} = \begin{pmatrix} 1 & 0.01 \\ -0.01 & 0.99 \end{pmatrix}$

Fixpoint computation

We write $\underline{x}(x,v)=x$, $\underline{v}(x,v)=v$, $\underline{E}(x,v)=2x^2+3v^2+2xv$ and $P=\{\underline{x},-\underline{x},\underline{v},-\underline{v},\underline{E}\}$.
For harmonic oscillator, the relaxed semantic $F^{\mathcal{R}}$:

$$
\begin{array}{rcl}
F_1^{\mathcal{R}}(w)(p) & = & \{0\leq \underline{x}(x,v)\leq 1,\, 0\leq \underline{v}(x,v)\leq 1,\, \underline{E}(x,v)\leq 7\} \\
F_2^{\mathcal{R}}(w)(p) & = & \sup\{w_1(p),w_3(p)\} \\
F_3^{\mathcal{R}}(w)(p) & = & \inf\limits_{\lambda\in\mathcal{F}(P,\mathbb{R}_+)} \sup\limits_{(x,v)} \sum\limits_{q\in P}\lambda(q)(w_2(q)-q(x))
\end{array}
$$

To compute the least fixpoint of $F^{\mathcal{R}}$, we use classically Kleene iteration.
With widening, Kleene iteration returns:
$\{-2.45\leq x\leq 2.45,\ -2\leq v\leq 2,\ 2x^2+3v^2+2xv\leq 10\}$.

Policy iteration

We remark that:

- $F_3^{\mathcal{R}}(w)(p) = \inf_{\lambda \in \mathcal{F}(P,\mathbb{R}_+)} \sup_{(x,v)} \sum_{q \in P} \lambda(q)(w_2(q) - q(x))$
- And for technical reasons, when we fix w^*, there exists a Lagrange multiplier λ^* s.t $F_3^{\mathcal{R}}(w) = \sup_{(x,v)} \sum_{q \in P} \lambda^*(q)(w_2^*(q) - q(x))$.

This two remarks allows us to use policy iteration [CGG+05, GGTZ07].

The policies are the set on which we minimize so, here, policies are the vectors of Lagrange multipliers.

Policy iteration

We remark that:

- $F_3^{\mathcal{R}}(w)(p) = \inf_{\lambda \in \mathcal{F}(P,\mathbb{R}_+)} \sup_{(x,v)} \sum_{q \in P} \lambda(q)(w_2(q) - q(x))$
- And for technical reasons, when we fix w^*, there exists a Lagrange multiplier λ^* s.t $F_3^{\mathcal{R}}(w) = \sup_{(x,v)} \sum_{q \in P} \lambda^*(q)(w_2^*(q) - q(x))$.

This two remarks allows us to use policy iteration [CGG+05, GGTZ07].

The policies are the set on which we minimize so, here, policies are the vectors of Lagrange multipliers.

1. First, PI consists in fixing a minimum i.e we fix a policy λ.
   For instance, $p = \underline{E}$, we can choose
   $\lambda(\underline{x}) = \lambda(-\underline{x}) = \lambda(\underline{v}) = \lambda(-\underline{v}) = 0$ and $\lambda(\underline{E}) = 1$. We get
   $(F_3^{\lambda}(w))(p) = w_2(\underline{E}) + \sup_{(x,v)} p(M(x,v)) - \underline{E}(x,v)$.
2. Then, we compute the LFP, w^{λ} of F^{λ} by solving a linear program.
3. Finally, we test if w^{λ} is a fixpoint of $F^{\mathcal{R}}$, if not, we change the policy.

Policy iteration

Howard (1-player game 60) Hoffman and Karp (Stochastic Games 66).

Costan,Gaubert,Goubault,Martel,Putot (2005) for Static analysis.

For $f := \inf_{\pi\in\Pi} f^{\pi}$ over a lattice L.

Assume that $\forall\, x \in L,\ \exists\, \pi$ s.t $f(x) = f^{\pi}(x)$. e.g: the set Π is finite.

Policy iteration (PI):

1. Select an initial policy π^0, $k = 0$
2. Compute the least fixpoint x^k of f^{π^k}
3. If $f(x^k) = x^k$ STOP else select π^{k+1} s.t $f(x^k) = f^{\pi^{k+1}}(x^k)$ go to 2.

Here

- The lattice L is the set $\mathcal{F}(P, \overline{\mathbb{R}})$.
- The function f is $F^{\mathcal{R}}$.
- Our policies are Lagrange multipliers.

Step 1

For all p, we take $\lambda(\underline{x}) = \lambda(-\underline{x}) = \lambda(\underline{v}) = \lambda(-\underline{v}) = 0$ and $\lambda(\underline{E}) = 1$ as initial policy.

We have to compute the least fixpoint of the "policy":

$$\begin{array}{rcl}(F_1^\lambda(w))(p) & = & \{0 \leq \underline{x}(x,v) \leq 1,\, 0 \leq \underline{v}(x,v) \leq 1,\, \underline{E}(x,v) \leq 7\} \\ (F_2^\lambda(w))(p) & = & \sup\{w_1(p), w_3(p)\} \\ (F_3^\lambda(w))(p) & = & w_2(\underline{E}) + \sup_{(x,v)} p(Mx) - \underline{E}(x,v)\end{array}$$

Step 2

we compute it by solving the linear program:

$$\text{Minimize } \sum_{i=1}^{3} \sum_{p\in P} w_i(p)$$

$$\begin{array}{c}(F_1^\lambda(w)) \leq w_1 \\ (F_2^\lambda(w)) \leq w_2 \\ (F_3^\lambda(w)) \leq w_3\end{array}$$

For instance, after a cloture operation, at the third component:

$$\begin{array}{rl} w_3^0(\underline{x}) = & 2.0493 \\ w_3^0(-\underline{x}) = & 2.0493 \\ w_3^0(\underline{v}) = & 1.6733 \\ w_3^0(-\underline{v}) = & 1.6733 \\ w_3^0(\underline{E}) = & 7.0000 \end{array}$$

Step 3

We evaluate $F^{\mathcal{R}}(w)$ using Shor relaxation.
We find $F^{\mathcal{R}}(w) \neq w$ but
Shor relaxation provides us optimal Lagrange multipliers which we use as policies.

At each $p \in P$, the policy $\pi_3^1(p)$ is the vector of Lagrange multipliers: $(\lambda(\underline{x}), \lambda(-\underline{x}), \lambda(\underline{v}), \lambda(-\underline{v}), \lambda(\underline{E}))$.

For instance: $\pi_3^1(\underline{x}) = (0.9035, 0, 0, 0, 0.0134)$.

Go back to **Step 2**.

Finally, after 5 iterations, the invariant of the loop i.e. $\gamma(w_2)$ at control point 2 is:

$$\{-1.8708 \leq x \leq 1.8708,\ -1.5275 \leq v \leq 1.5275,\ 2x^2 + 3v^2 + 2xv \leq 7\}.$$

Programs	M	#P	#lin	#var	#lps	#Ite.	Qual	Tim
Rotation2	P	2	2	2	0	0	FP	0.72
Rotation2	K	2	2	2	0	1	FP	1.07
Rotation10	P	2	2	10	0	0	FP	1.17
Rotation10	K	2	2	10	0	1	FP	1.82
Filter	P	5	3	2	1	2	FP	9.35
Filter	K	5	3	2	1	2	FP	19.7
Oscillator	P	5	3	2	1	5	FP	12
Oscillator	K	5	3	2	1	15	FP	18.8
Symplectic	P	5	3	2	1	0	FP	3
Symplectic	K	5	3	2	1	15	FP	18.3
SymplecticSeu	P	5	5	2	1	30	PFP	125.3
SymplecticSeu	K	5	5	2	1	30	PFP	78.9
Arrow-Hurwicz	P	2	14	4	3	10	PFP	44.6
Arrow-Hurwicz	K	2	14	4	3	26	PFP	81.7

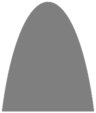
Quality of invariant is very good with PI.

Programs	M	#P	#lin	#var	#lps	#Ite.	Qual	Tim
Rotation2	P	2	2	2	0	0	FP	0.72
Rotation2	K	2	2	2	0	1	FP	1.07
Rotation10	P	2	2	10	0	0	FP	1.17
Rotation10	K	2	2	10	0	1	FP	1.82
Filter	P	5	3	2	1	2	FP	9.35
Filter	K	5	3	2	1	2	FP	19.7
Oscillator	P	5	3	2	1	5	FP	12
Oscillator	K	5	3	2	1	15	FP	18.8
Symplectic	P	5	3	2	1	0	FP	3
Symplectic	K	5	3	2	1	15	FP	18.3
SymplecticSeu	P	5	5	2	1	30	PFP	125.3
SymplecticSeu	K	5	5	2	1	30	PFP	78.9
Arrow-Hurwicz	P	2	14	4	3	10	PFP	44.6
Arrow-Hurwicz	K	2	14	4	3	26	PFP	81.7

Time is quite big because the prototype is Matlab code (interpreted code). But a small time is passed in solvers (0.2 seconds).

Programs	M	#P	#lin	#var	#lps	#Ite.	Qual	Tim
Rotation2	P	2	2	2	0	0	FP	0.72
Rotation2	K	2	2	2	0	1	FP	1.07
Rotation10	P	2	2	10	0	0	FP	1.17
Rotation10	K	2	2	10	0	1	FP	1.82
Filter	P	5	3	2	1	2	FP	9.35
Filter	K	5	3	2	1	2	FP	19.7
Oscillator	P	5	3	2	1	5	FP	12
Oscillator	K	5	3	2	1	15	FP	18.8
Symplectic	P	5	3	2	1	0	FP	3
Symplectic	K	5	3	2	1	15	FP	18.3
SymplecticSeu	P	5	5	2	1	30	PFP	125.3
SymplecticSeu	K	5	5	2	1	30	PFP	78.9
Arrow-Hurwicz	P	2	14	4	3	10	PFP	44.6
Arrow-Hurwicz	K	2	14	4	3	26	PFP	81.7

```
x = [0,1];
v = [0,1];
tau = 0.1 [1]
while [2] ((v>=1/2) [3]) { [4]
  x = (1-tau/2)*x+(tau-(tau^3)/4)*v;
  v = -tau*x+(1-tau/2)*v; [5]};
```



$$\left\{\begin{array}{c} -1.049 \leq x \leq 1.049 \\ -0.5 \leq v \leq 1.163 \\ x^2 + 0.9975 v^2 \leq 1.35 \end{array}\right\}$$

Figure: The symplectic implementation code with a threshold

Contributions and future works

Contributions

- New Domain for abstract interpretation.
- Generalization of linear templates Manna et al [SSM05, SCSM06]: To find tigher bounds on variables, we use nonlinear templates.
- We can compute an overapproximation of our abstract semantic in polynomial time.
- Policy iteration algorithm [CGG$^+$05, GGTZ07].

Future work

Consider programs with polynomial arithmetic and deal with sum-of-squares (SOS) relaxations.

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 *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *LNCS*, pages 462–475. Springer, 2005.

S. Gaubert, E. Goubault, A. Taly, and S. Zennou.
Static analysis by policy iteration on relational domains.
In *Proceedings of the Sixteenth European Symposium Of Programming (ESOP'07)*, volume 4421 of *LNCS*, pages 237–252. Springer, 2007.

Y. Nesterov and A. Nemirovski.
Interior point polynomial algorithms in convex programming.
Society for Industrial and Applied Mathematics, 1994.

M. Pardalos and S. A. Vavasis.
Quadratic programming with one negative eigenvalue is np-hard.
Journal of Global Optimization, 1(1):15–22, March 1991.

S. Sankaranarayanan, M. Colon, H. Sipma, and Z. Manna.
Efficient strongly relational polyhedral analysis.
In E. Allen Emerson and Kedar S. Namjoshi, editors, *Verification, Model Checking, and Abstract Interpretation: 7th International Conference, (VMCAI)*, volume 3855 of *LNCS*, pages 111–125, Charleston, SC, January 2006. Springer.

N. Shor.
Quadratic optimization problems.
Soviet J. of Computer and Systems Science, 25(6):1–11, 1987.

S. Sankaranarayanan, H. B. Sipma, and Z. Manna.
Scalable analysis of linear systems using mathematical programming.
In *Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, volume 3385 of *LNCS*, pages 25–41, January 2005.

A. Ben Tal and A. Nemirowski.
Lecture on Modern Convex Optimization: Analysis, Algorithm and Engineering Applications.

SIAM, 2001.