Problems of optimisation and game theory in static analysis of programs

Stéphane Gaubert
INRIA

Stephane.Gaubert@inria.fr

OPTIMEO day, Palaiseau, April 4 2008

Joint work with: Éric Goubault, Sylvie Putot, Assale Adje (CEA/MeASI) and Xavier Allamigeon (EADS)

How to prove that ?

```
void main() {
 i = 1; j = 10;
 while (i <= j){ //1
  i = i + 2;
  j = j - 1; }
}
```

$$i \leq +\infty$$
$$i \geq 1$$
$$j \leq 10$$
$$j \geq -\infty$$
$$i \leq j$$
$$i + 2j \leq 21$$
$$i + 2j \geq 21$$

$(i, j) \in [(1, 10), (7, 7)]$ (exact result).

A possible implementation of the C standard library function `memcpy`

```
int i := 0;
unsigned int n, p, q;
string dst[p], src[q];
assert p >= n && q >= n;
while i <= n-1 do
  dst[i] := src[i];
  i := i+1;
done;
```

How to prove that ?

$$\min(\texttt{len\_src}, \texttt{n}) = \min(\texttt{len\_dst}, \texttt{n})$$

Bubble sort

```
Variables: i, j, k, x, y, z

Program:
local t {
i:=x;
j:=y;
k:=z;
if x > y then
  i:=y;
  j:=x;
fi;
if j > z then
  k:=j;
  j:=z;
fi;
if i > j then
  t:=j;
  j:=i;
  i:=t;
fi;
};
```

How to prove that ?

```
k = max(x,y,z);
```

or even that. . .

```
-y = max(-k,-y); max(-k,-z) = -z; max(-j,-x,-z) = max(-x,-z);
  -j = max(-j,-k); max(-y,-z) = max(-j,-y,-z); max(j,y,z) = max(y,z);
  z = max(i,z); -x = max(-k,-x); max(-x,-y) = max(-j,-x,-y); -i = max(-i,-x);
  max(-x,-y,-z) = max(-i,-k); x = max(i,x); max(j,x,z) = max(x,z);
  max(i,y) = y; max(j,x,y) = max(x,y); j = max(i,j); k = max(x,y,z)
```

Answer:

convex analysis (including **generalized convexity**)

and **zero-sum games**

Since Cousot and Halbwachs (POPL'78), polyhedra have been used in static analysis by abstract interpretation:

show that for any reachable state of the program, the vector consisting of the variables at the different breakpoints belongs to a polyhedron.

repeatedly perform some basic operations: intersection, convex hull (e.g. of union), image by an affine map

strongly relies on **convex duality**

BUT the number of extreme points or faces may grow exponentially

$\longrightarrow$ not scalable

Some restricted classes of polyhedra have been introduced. Miné (PADO'01) used Zones

$Z = \{x \in \mathbb{R}^n \mid \; x_i - x_j \leq M_{ij}\}$

a zone is coded by the matrix $M \in (\mathbb{R} \cup \{+\infty\})^{n \times n}$.

by setting $x_0 := 0$ and projecting, we see that Zones $\supset$ Intervals.

S. Sankaranarayanan and H. Sipma and Z. Manna (VMCAI'05) introduced templates:

almost as expressive as polyhedra but scalable.

I'll give a convex analytic view of templates.

The **support function** σ_X of $X \subset \mathbb{R}^n$ is defined by

$\sigma_X(p) = \sup_{x \in X} p \cdot x$

Legendre-Fenchel duality tells that $\sigma_X = \sigma_Y$ iff X and Y have the same closed convex hull.

$\sigma_X(\alpha p) = \alpha \sigma_X(p)$ for $\alpha > 0$, so it is enough to know $\sigma_X(p)$ for all p in the unit sphere.

Idea: discretize the unit sphere and represent X by σ_X restricted to the discretization points.

So fix $\mathcal{P} \subset \mathbb{R}^n$ a finite set of directions.

$L(\mathcal{P})$ lattice of sets of the form

$Z = \{x \mid p \cdot x \leq \gamma(p), \; \forall p \in \mathcal{P}\}$, $\quad \gamma : \mathcal{P} \to \mathbb{R} \cup \{+\infty\}$.

Z is coded by $\gamma := \sigma_Z \mid_{\mathcal{P}}$.

Z is a polyhedron every facet of which is orthogonal to some $p \in \mathcal{P}$.

Specialization: $\mathcal{P} = \{\pm e_i, \; i = 1, \ldots, n\}$ gives intervals, $\mathcal{P} = \{\pm(e_i - e_j), \; 1 \leq i < j \leq n\}$ gives Miné's templates.

```
void main() {
 i = 1; j = 10;
 while (i <= j){ //1
  i = i + 2;
  j = j - 1; }
}
```

$$i \leq +\infty$$

$$i \geq 1$$

$$j \leq 10$$

$$j \geq -\infty$$

$$i \leq j$$

$$i + 2j \leq 21$$

$$i + 2j \geq 21$$

```
void main() {
 i = 1; j = 10;
 while (i <= j){ //1
  i = i + 2;
  j = j - 1; }
}
```

$$
\begin{aligned}
\gamma(e_1) &= +\infty \\
\gamma(-e_1) &= -1 \\
\gamma(e_2) &= 10 \\
\gamma(-e_2) &= -\infty \\
\gamma(e_1 - e_2) &= 0 \\
\gamma(e_1 + 2e_2) &= 21 \\
\gamma(-e_1 - 2e_2) &= -21 \ .
\end{aligned}
$$

$\mathcal{P} = \{\pm e_1, \pm e_2, e_1 - e_2, \pm(e_1 + 2e_2)\}$, γ: breakpoint 1.

To show this, we must solve the fixed point problem:

$$\gamma(p) = ((1,10) \cdot p) \vee (\bar{\gamma}(p) + (2,-1) \cdot p), \ \forall p \in \mathcal{P} \setminus \{e_1 - e_2\}$$

$$\gamma(e_1 - e_2) = 0 \wedge (-9 \vee (\bar{\gamma}(e_1 - e_2) - 3)), \quad \bar{\gamma} = \text{convex hull}(\gamma)$$

```
void main() {
 i = 1; j = 10;
 while (i <= j){ //1
  i = i + 2;
  j = j - 1; }
}
```

Correspondence theorem (SG, Goubault, Taly, Zennou, ESOP'07) When the arithmetics of the program is *affine* (no product or division of variables), abstract interpretation over a lattice of templates reduces to finding the smallest fixed point of a map $f : (\mathbb{R} \cup \{+\infty\})^n \to (\mathbb{R} \cup \{+\infty\})^n$ of the form

$$f_i(x) = \inf_{a \in A(i)} \sup_{b \in B(i,a)} (r_i^{ab} + M_i^{ab} x)$$

with $M_i^{ab} := (M_{ij}^{ab})$, $M_{ij}^{ab} \geq 0$, but possibly $\sum_j M_{ij}^{ab} > 1$

$\rightarrow$ game in infinite horizon with a "negative discount rate".

Sketch of proof.

$y = Ax + b$; If $x \in Z^1 := \{z \mid p \cdot z \leq \gamma^1(z),\ \forall p \in \mathcal{P}\}$, find the best $Z^2 := \{z \mid p \cdot z \leq \gamma^2(z),\ \forall p \in \mathcal{P}\}$ such that $y \in Z^2$.

$\gamma^2(p) = \sup_{x \in Z^1} p \cdot (Ax + b) = \sup p \cdot (Ax + b);\ p \cdot x \leq \gamma^1(p),\ \forall p \in \mathcal{P}$

by the strong duality theorem

$= \inf p \cdot b + \sum_{q \in \mathcal{P}} \lambda(q) \gamma^1(q); \quad \lambda(q) \geq 0, \quad A^T p = \sum_{q \in \mathcal{P}} \lambda(q) q$

The inf is attained at an extreme point of the feasible set, so this is in fact a min over a finite set.

$\sigma_{X \cap Y} =$ convex hull$(\inf(\sigma_X, \sigma_Y))$.

Convex hull reduces to a finite min by a similar argument.

Modelling the dataflow yields maxima, because $\sigma_{X \cup Y} = \sup(\sigma_X, \sigma_Y)$

Generalization of templates (Adje, SG, Goubault, Putot, current investigation):

$$Z = \{x \mid p(x) \leq \gamma(p),\ \forall p \in \mathcal{P}\}$$

p is now a non-linear map (e.g. quadratic, e.g. Lyapunov function). The fixed point operator involves SDP relaxations (could even use SOS).

How to solve the fixed point problem ?

Classically: Kleene (fixed point iteration) is slow or may even not converge, so widening and narrowing have been used, leading to an overapproximation of the solution.

An alternative: **Policy iteration.**

method developed by Howard (60) in stochastic control, extended by Hofman and Karp (66) to some special (nondegenerate) stochastic games. Extension to Newton method $\Longrightarrow$ fast. complexity still open.

extended by Costan, SG, Goubault, Martel, Putot, CAV'05) to fixed point problems in static analysis (difficulty: what are the strategies?)

experiments: PI often yields more accurate fixed points (because it avoids widening), small number of iterations.

A **strategy** is a map π which to a state i associates an action $\pi(i) \in A(i)$.

Consider the one player dynamic programming operator:

$$f_i^\pi(x) := \sup_{b \in B(i,\pi(i))} (r_i^{\pi(i)b} + M_i^{\pi(i)b} x)$$

$$f = \inf_\pi f^\pi$$

and the set $\{ f^\pi \mid \pi$ strategy $\}$ has a *selection*:

$$\forall v \in \mathbb{R}^n,\ \exists \pi \qquad f(v) = f^\pi(v) \enspace .$$

Since f^π is convex and piecewise affine, finding the smallest finite fixed point of f^π (if any) can be done by linear programming:

$$\min \sum_i v_i; \quad f^\pi(v) \leq v \enspace .$$

Costan, SG, Goubault, Martel, Putot (CAV'05) show that the smallest fixed point of f is the infimum of the smallest fixed points of the f^{π}.

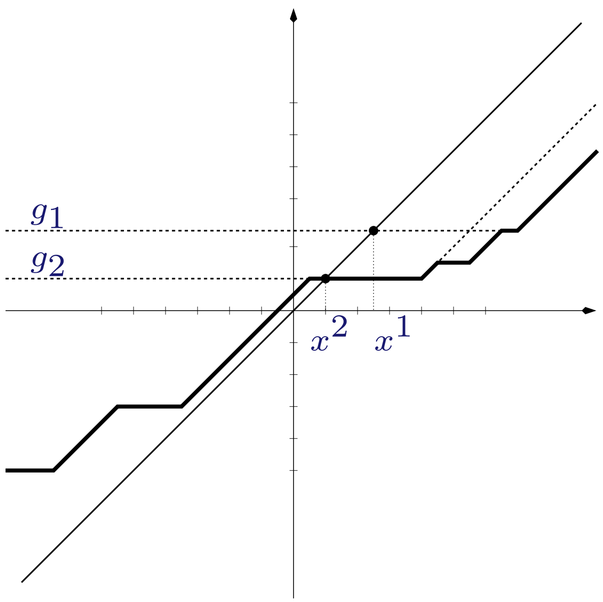
We denote by f^{-} the smallest fixed point of a monotone self-map f of a complete lattice $\mathcal{L}$, whose existence is guaranteed by Tarski's fixed point theorem.

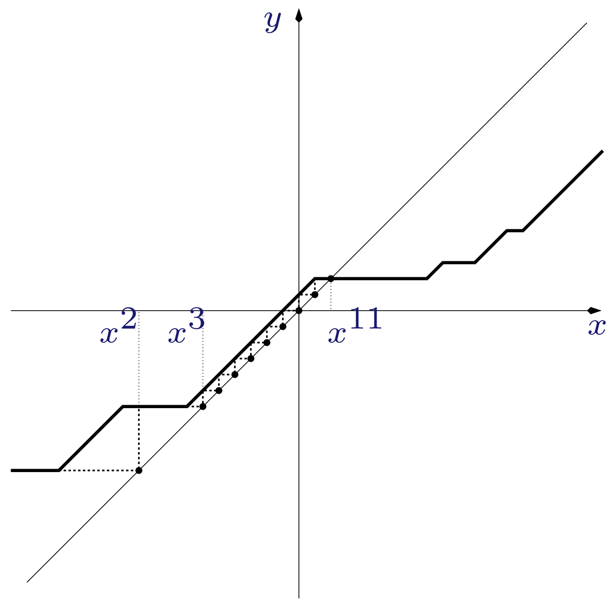
The input of the following algorithm consists of a finite set $\mathcal{G}$ of monotone self-maps of a lattice $\mathcal{L}$ with a lower selection. When the algorithm terminates, its output is a fixed point of $f = \inf \mathcal{G}$.

1. *Initialization*. Set $k = 1$ and select any map $g_1 \in \mathcal{G}$.
2. *Value determination*. Compute a fixed point x^k of g_k.
3. Compute $f(x^k)$.
4. If $f(x^k) = x^k$, return x^k.
5. *Policy improvement*. Take g_{k+1} such that $f(x^k) = g_{k+1}(x^k)$. Increment k and goto Step 2.

The algorithm does terminate when at each step, the smallest fixed-point of g_k, $x^k = g_k^-$ is selected.

Example. Take $\mathcal{L} = \overline{\mathbb{R}}$, and consider the self-map of $\mathcal{L}$, $f(x) = \inf_{1\leq i\leq m} \max(a_i + x, b_i)$, where $a_i, b_i \in \mathbb{R}$. The set $\mathcal{G}$ consisting of the m maps $x \mapsto \max(a_i + x, b_i)$ admits a lower selection.





Experimentally fast, but the worst case complexity is not known. Condon showed: mean payoff games is in NP $\cap$ co-NP, same with positive discount. Much current work: (Zwick, Paterson, TCS 96), (Jurdziński, Paterson, Zwick, SODA'06), (Bjorklund, Sandberg, Vorobyov, preprint 04),

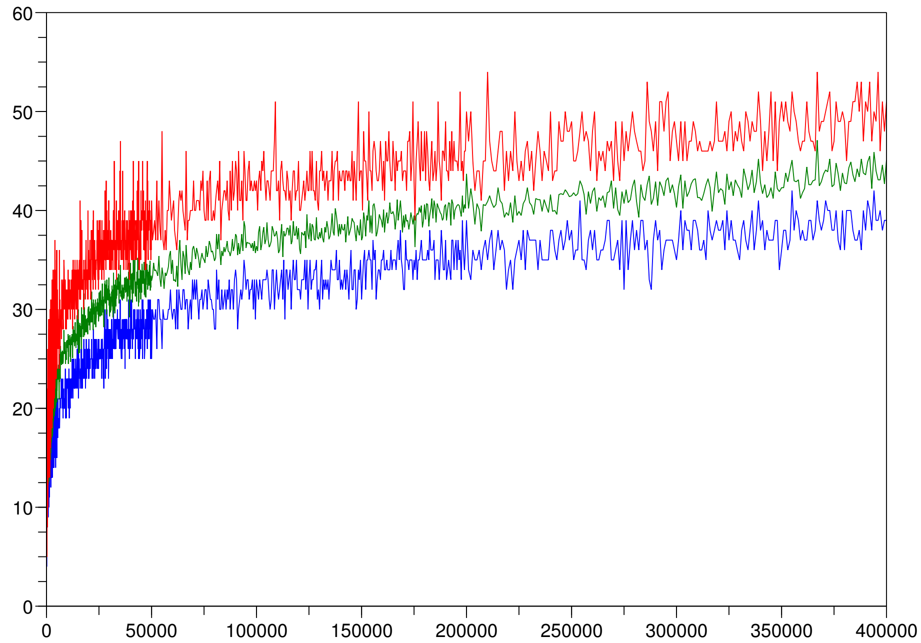
PI often more accurate than Klenne+widening/narrowwing:

```
0      i = 150;
1      j = 175;
2
 while (j >= 100){
3          i++;
4          if (j<= i){
5
 i = i - 1;
6
 j = j - 2;
7          }
8     }
9
```

$$
\begin{array}{lcl}
M_0 & = & \mathit{context\_initialization} \\
M_2 & = & (\mathit{Assignment}\ (i \leftarrow 150,\ \ j \leftarrow 175)(M_0))^* \\
M_3 & = & ((M_2 \sqcup M_8) \sqcap (j \geq 100))^* \\
M_4 & = & (\mathit{Assignment}\ (i \leftarrow i + 1)(M_3))^* \\
M_5 & = & (M_4 \sqcap (j \leq i))^* \\
 & & \\
M_7 & = & (\mathit{Assignment}\ (i \leftarrow i - 1,\ \ j \leftarrow j - 2)(M_5))^* \\
M_8 & = & ((M_4 \sqcap (j > i))^* \sqcup M_7 \\
M_9 & = & ((M_2 \sqcup M_8) \sqcap (j < 100))^*
\end{array}
$$

IP $\left\{ \begin{array}{c} 150 \leq i \leq 174 \\ 98 \leq j \leq 99 \\ -76 \leq j - i \leq -51 \end{array} \right.$ Mine's Octogon $\left\{ \begin{array}{c} 150 \leq i \\ 98 \leq j \leq 99 \\ j - i \leq -51 \\ 248 \leq j + i \end{array} \right.$

SG, Dhingra (Valuetools'06). Sparse bipartite graphs. n nodes of each kind, every node has exactly 2 successors drawn at random; , deterministic game, random weights. Number of iterations of minimizer $N_{\min}$ is shown:



Difficulty

PI may return a nonminimal fixed point.

We know there is a policy yielding the minimal fixed point.

How to find it?

Theorem. Adje, SG, Goubault (MTNS'08, to appear).

If f is nonexpansive (1-Lip) in the sup-norm, i.e., if there is no negative discount rate, we can refine PI so that it always finds the smallest fixed point.

Relies on: in finite dimension, the fixed point set of a nonexpansive map is a retract of the whole space.

If negative discount is allowed, the fixed point set may be disconnected, we can always reach a locally minimal fixed point. . .

finding efficiently the globally minimal one is an open question.

```
int x,int y,
x=[0,2];y=[10,15]     //1
while (x<=y) {        //2
  x=x+1;              //3
  while (5<=y) {      //4
    y=y-1;            //5
  }                   //6
}                     //7
```

$$
\begin{aligned}
(x_1, y_1) &= ([0,2],[10,15]) \\
x_2 &= (x_1 \cup x_6) \cap [-\infty, (y_1 \cup y_6)^+] \\
y_2 &= (y_1 \cup y_6) \cap [(x_1 \cup x_6)^-, +\infty] \\
(x_3, y_3) &= (x_2 + [1,1], y_2) \\
(x_4, y_4) &= (x_3, (y_3 \cup y_5) \cap [5, +\infty]) \\
(x_5, y_5) &= (x_4, y_4 + [-1,-1]) \\
(x_6, y_6) &= (x_5, (y_3 \cup y_5) \cap [-\infty, 4]) \\
x_7 &= (x_1 \cup x_6) \cap [(y_1 \cup y_6)^- + 1, +\infty] \\
y_7 &= (y_1 \cup y_6) \cap [-\infty, (x_1 \cup x_6)^+ - 1]
\end{aligned}
$$

The monotone nonexpansive piecewise affine map f for the bounds of these intervals is:

$$f\begin{pmatrix} x \\ y \end{pmatrix} = f\begin{pmatrix} x_2^- \\ x_2^+ \\ x_7^- \\ x_7^+ \\ y_2^- \\ y_2^+ \\ y_4^- \\ y_4^+ \\ y_6^- \\ y_6^+ \\ y_7^- \\ y_7^+ \end{pmatrix} = \begin{pmatrix} 0 & \vee & (x_2^- - 1) \\ 2 \vee (x_2^+ + 1) & \wedge & \underline{15 \vee y_6^+} \\ 0 \vee (x_2^- - 1) & \wedge & \underline{(-10 \vee y_6^-) - 1} \\ 0 & \vee & (x_2^+ + 1) \\ \underline{0 \vee (x_2^- - 1)} & \wedge & -10 \vee y_6^- \\ 15 & \vee & y_6^+ \\ y_2^- \vee (y_4^- + 1) & \wedge & \underline{-5} \\ y_2^+ & \vee & y_4^+ - 1 \\ y_2^- & \vee & y_4^- + 1 \\ y_2^+ \vee (y_4^+ - 1) & \wedge & \underline{4} \\ -10 & \vee & y_6^- \\ 15 \vee y_6^+ & \wedge & \underline{(2 \vee (x_2^+ + 1)) - 1} \end{pmatrix}$$

The underlined terms represent the initial Policy. We find $(\bar{x}, \bar{y}) = (0, 15, -1, 16, 0, 15, -5, 15, 0, 4, 0, 15)$: it is a fixed point of f, and so policy iteration terminates in one step.

We calculate the semidifferential at $(\bar{x}, \bar{y})$ in the direction $(\delta x, \delta y)$.

$$f'_{(\bar{x},\bar{y})}(\delta\bar{x},\delta\bar{y}) \quad = \quad \Big(0,0,\delta\bar{x}_2^- \wedge \delta\bar{y}_6^-, \delta\bar{x}_2^+, 0 \wedge \delta\bar{y}_6^-, 0, 0, \delta\bar{y}_2^+, \delta\bar{y}_2^-, 0, \delta\bar{y}_6^-, 0 \wedge \delta\bar{x}_2^+, \Big)$$

The power algorithm gives us $h = (0,0,-1,0,-1,0,0,0,-1,0,-1,0)$ (computed from the iterates of the vector with all coordinates equal to -1). We know that there is an integer $t < 0$ such that $(\bar{x},\bar{y}) - th$ is a fixed point of f.

$$f((\bar{x},\bar{y}) - th) = f\begin{pmatrix} & 0 & \\ & 15 & \\ -1 & + & t \\ & 16 & \\ & t & \\ & 15 & \\ & -5 & \\ & 15 & \\ & t & \\ & 4 & \\ & t & \\ & 15 & \end{pmatrix} = \begin{pmatrix} & 0 & \\ & 15 & \\ 0 & \wedge & (-10 \vee t) - 1 \\ & 16 & \\ 0 & \wedge & -10 \vee t \\ & 15 & \\ t \vee -4 & \wedge & -5 \\ & 15 & \\ t & \vee & -4 \\ & 4 & \\ -10 & \vee & t \\ & 15 & \end{pmatrix}$$

The smallest such t is -4. We find a new fixed point $(\tilde{u}, \tilde{v}) = (0, 15, -5, 16, -4, 15, -5, 15, -4, 4, -4, 15)$ for f. The semidifferential at $(\tilde{u}, \tilde{v})$ is then:

$$f'_{(\tilde{u},\tilde{v})}(\delta\tilde{u}, \delta\tilde{v}) = \left(0, 0, \delta\tilde{v}_6^-, \delta\tilde{u}_2^+, \delta\tilde{v}_6^-, 0, 0, \delta\tilde{v}_2^+, \delta\tilde{v}_2^- \vee \delta\tilde{v}_4^-, 0, \delta\tilde{v}_6^-, 0 \wedge \delta\tilde{u}_2^+\right)$$

The power algorithm returns 0 (again with iterates of the vector identically equal to -1) , we conclude that $(\bar{x}, \bar{y})$ is the smallest fixed point of f.

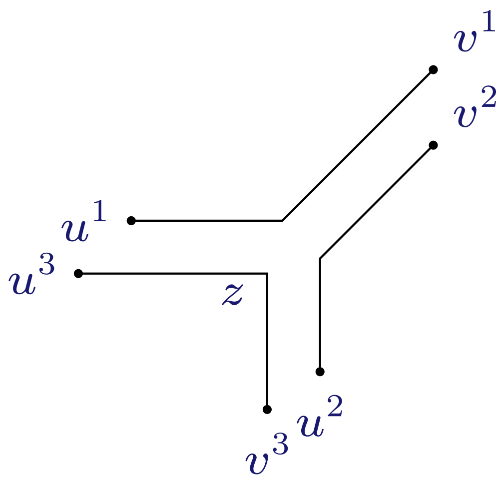
Exotic domains in static analysis . . .

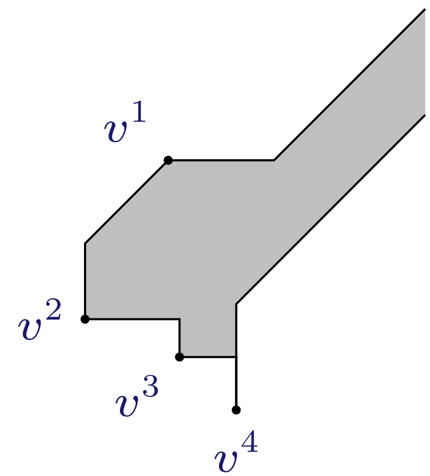
max-plus or tropical convex sets

(Allamigeon, SG, Goubault, SAS'08 to appear)

A subset C of $(\mathbb{R} \cup \{-\infty\})^n$ is max-plus convex if

$$x, y \in C,\ \lambda, \mu \in \mathbb{R} \implies \sup(\lambda + x, \mu + y) \in C \ .$$

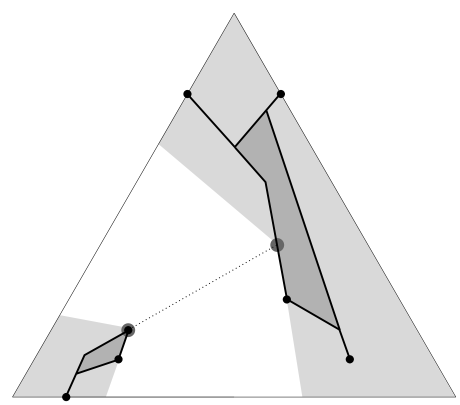




Considered by U. Zimmermann (77), Cohen, SG, Quadrat (00), Sturmfels, Develin (04), +recent: Katz, Horvath, Sergeev, Meunier, . . .

Separation theorem, projection, minimisation of distance, discrete convexity (Helly, Carathéodory), Minkowski, Krein-Milman, or Choquet theory (generation by extreme points) carry over.

Ex. Separation of two convex sets, SG & Sergeev (07):



A max-plus polyhedron is the sum of a max-plus polytope and a max-plus polyhedral cone, or equivalently, the intersection of finitely many half-spaces

$$H = \{x \mid \max_i a_i + x_i \leq \max_i b_i + x_i\} \ .$$

Fourier-Motzkin type algorithms work.

As for classical polyhedra, passing from generators to constraints and vice versa is simply exponential.

In Allamigeon, SG, Goubault (SAS'08, to appear), we handle max-plus polyhedra coded by constraints: Kleene iteration with Cousot's widening. This is how we got:

```
Variables: i, j, k, x, y, z

Program:
local t {
i:=x;
j:=y;
k:=z;
if x > y then
  i:=y;
  j:=x;
fi;
if j > z then
  k:=j;
  j:=z;
fi;
if i > j then
  t:=j;
  j:=i;
  i:=t;
fi;
};
```

```
-y = max(-k,-y); max(-k,-z) = -z; max(-j,-x,-z) = max(-x,-z);
  -j = max(-j,-k); max(-y,-z) = max(-j,-y,-z); max(j,y,z) = max(y,z);
  z = max(i,z); -x = max(-k,-x); max(-x,-y) = max(-j,-x,-y); -i = max(-i,-x);
  max(-x,-y,-z) = max(-i,-k); x = max(i,x); max(j,x,z) = max(x,z);
```

```
max(i,y) = y; max(j,x,y) = max(x,y); j = max(i,j); k = max(x,y,z)
```

Concluding remarks

- Open complexity/algorithmic issue: smallest fixed point of a Shapley operator, with negative discount.

- Optimal complexity for handling max-plus polyhedra not yet known.

- General use of nonconvex domains in static analysis, with SDP or SOS relaxations: to be done (see already work by Feron, also current work by Monniaux).

That's all. . .