40 years of convex polyhedra, and what's more to say?

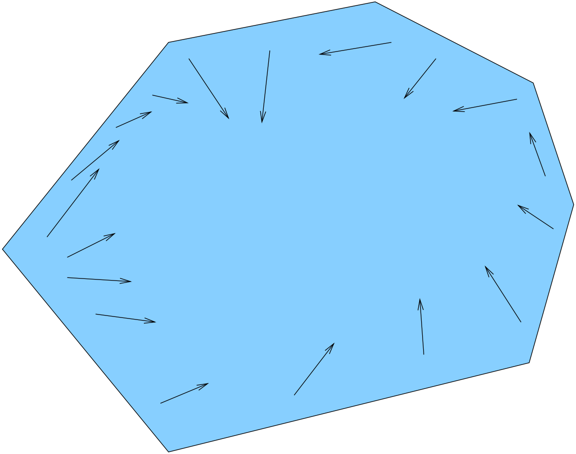
David Monniaux

VERIMAG

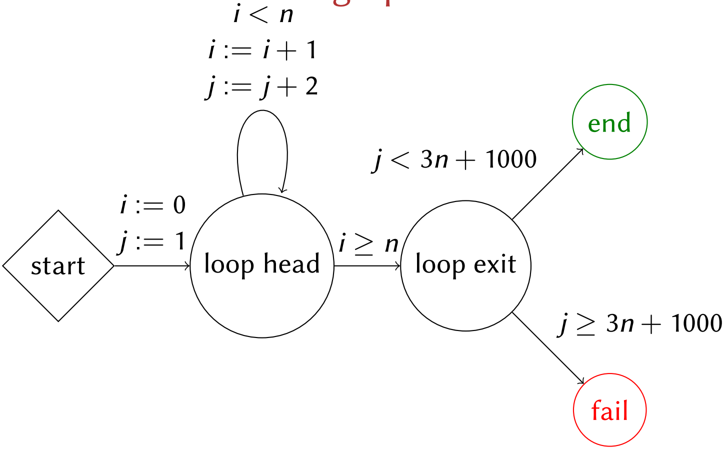
2018-06-04

Plan

Invariants for dynamic systems



Invariants for control-flow graphs



Cousot & Halbwachs, 1978

Halbwachs, 1979

Loop nests

```
for(int i=0; i<n; i++) {    // 0 ≤ i ≤ n
  for(int j=i; j<n; j++) { // 0 ≤ i ≤ j ≤ n
    t[i][j] = 42;
  }
}
```

Is there anything wrong?

Loop nests

```
for(int i=0; i<n; i++) {     // 0 ≤ i ≤ n
  for(int j=i; j<n; j++) { // 0 ≤ i ≤ j ≤ n
    t[i][j] = 42;
  }
}
```

Is there anything wrong?

Need to assume $n > 0$.

Loops

```
assume(n > 0);
i = 0; j = n;
while(i < j) { // 0 ≤ i ≤ j ≤ n ∧ i + j = n
  i++;
  j--;
}
```

Curse of dimensionality

Costs tend to increase exponentially with number of variables.

Plan

Polyhedra as invariants

Double description

Restricted polyhedra

Constraints only, Fourier-Motzkin

Our contributions

Why heuristics?

Double description

Generators

Convex hull of

- vertices
- rays
- lines

Constraints

Solution set of a system of

- inequalities
- equalities

Duality

constraints $\leftrightarrow$ generators

faces $\leftrightarrow$ vertices

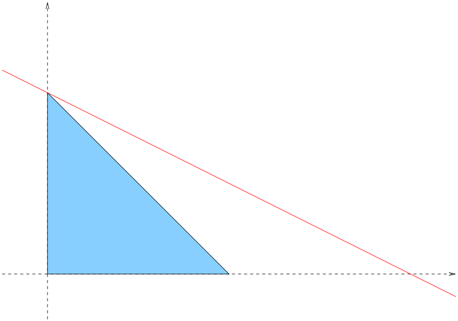
convex hull $\leftrightarrow$ intersection

inclusion $\leftrightarrow$ reverse inclusion

Any worst case on one description is a worst-case on the dual for a dual operation!

Redundancy of constraints

$$\begin{cases} x & \geq & 0 \\ y & \geq & 0 \\ x + y & \leq & 1 \\ x + 2y & \leq & 2 \end{cases}$$



The last constraint is **redundant**: all points satisfying the other constraints satisfy it.
It can be safely removed.

Witness of redundancy

$$\begin{array}{lrrcl} (1) & -x & & \leq & 0 \\ & & -y & \leq & 0 \\ (2) & x & +y & \leq & 1 \\ \hline & x & +2y & \leq & 2 \end{array}$$

Farkas lemma: semantic consequence $\Longleftrightarrow$
positive combination of original inequalities (plus slack)

Unicity of representation

If the polyhedron has **nonempty interior** (= is **not flat**)

Unique set of irredundant constraints
(up to scaling and rearranging: $2x - 2 \leq 0$ same as $x \leq 1$)

Each constraint defines a **true face** of the polyhedron.

Empty interior

No canonicity

$$\begin{cases} x & \leq & y+z \\ y+z & \leq & t \\ t & \leq & x \\ 0 & \leq & x \\ t & \leq & 1 \end{cases}$$

equivalent to

$$\begin{cases} x & \leq & y+z \\ y+z & \leq & t \\ t & \leq & x \\ 0 & \leq & x \\ x & \leq & 1 \end{cases}$$

Affine span

Extract a system of equalities defining the **affine span**

$$\begin{cases} x = y + z = t \\ 0 \leq x \\ t \leq 1 \end{cases}$$

Orient the equations of the affine span into a rewriting system (**variable ordering**: x, y function of z, t): $x \longrightarrow t$, $y \longrightarrow t - z$.
Canonify:

$$\begin{cases} x = y + z = t \\ 0 \leq t \leq 1 \end{cases}$$

Chernikova's algorithm

Step

Inputs: one polyhedron P as generators, one inequality I
Output: $P \cap I$ as generators

Constraints to generators

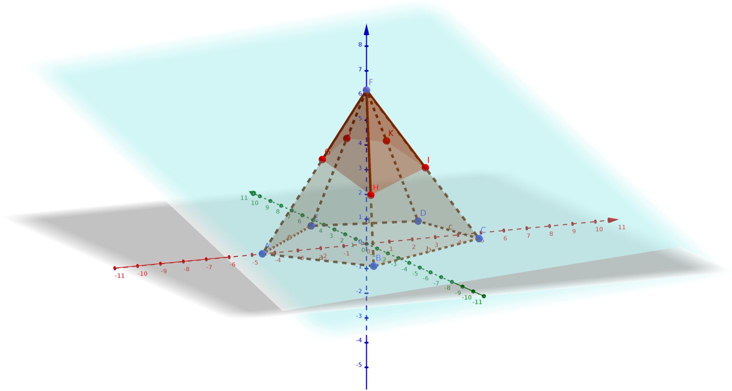
Process all constraints sequentially from full polyhedron

Generators to constraints

Dually

Le Verge, A Note on Chernikova's algorithm (1996)

Chernikova in action



Distorted hypercube

Very common in program analysis (known intervals).

$$\begin{cases} l_1 & \leq & x_1 & \leq & h_1 \\ \vdots & & \vdots & & \vdots \\ l_n & \leq & x_n & \leq & h_n \end{cases}$$

$2n$ constraints
2^n vertices

All libraries computing with double description explode.

Avoiding blowup

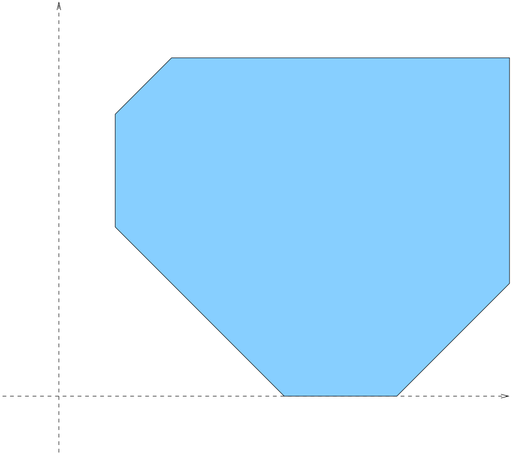
Halbwachs, Merchat, Gonnord (2006): factor polyhedra into products

Same principle in ETHZ's ELINA library (2017)

Our solution: constraints only

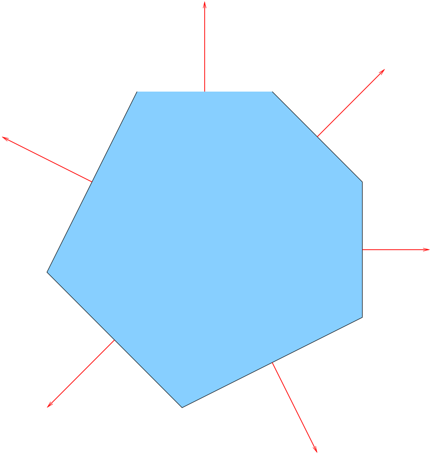
Plan

Octagons



system of $\pm v_1 \pm v_2 \leq C$ and $\pm v \leq C$

Templates



fixed set of normal vectors

Exact solving

Can solve for the least inductive invariant in a template linear constraint domain.

See as optimization (minimization) problem on the right-hand sides b.

"Does there exist an inductive invariant with b_i less than C?"

- Arbitrary polynomial arithmetic on the edges: reduction to $\exists\forall$ formula in real closed fields.
- Linear arithmetic, $\exists$, $\wedge$, $\vee$ on the edges: problem is Σ_p^2-complete.

Plan

Constraint-only representation

Easy

- intersection

Moderately easy

LP = linear programming, n = number of constraints

- emptiness check (1 LP)
- redundancy elimination (n LP)

How?

- projection
- convex hull

Fourier-Motzkin

$$\mathcal{S}\left\{\begin{array}{c} x \leq \dots \\ \vdots \\ x \leq \dots \\ \hline x \geq \dots \\ \vdots \\ \hline \text{not depending on } x \\ \vdots \\ \text{not depending on } x \end{array}\right.$$

- Combine each $x \leq \dots$ with each $x \geq \dots$:

$$f_1(y, z, \dots) \leq x \leq f_2(y, z, \dots) \longrightarrow f_1(y, z, \dots) \leq f_2(y, z, \dots)$$

- Keep the inequalities not depending on x.

Resulting system $\equiv \exists x\, \mathcal{S}$

Fourier-Motzkin

Pros

- Easy algorithm
- Easy proof of correctness (nice if doing Coq)

Cons

- Generates a huge volume of **redundant constraints**
  (Worst-case output $n^2/4$ for one projection.
  Can it actually go **double exponential** with number of projections if not removing redundancies?)
- If projecting several variables: chose an ordering on the canonical basis, not much geometrical.

Redundancy elimination by linear programming

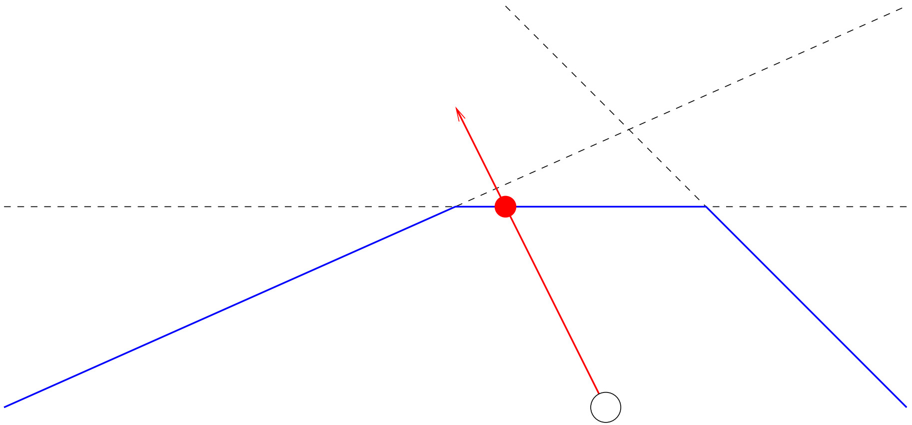
"Is C redundant with respect to $C_1 \wedge \cdots \wedge C_n$."

- **Primal** "Find x satisfying $C_1 \wedge \cdots \wedge C_n$ but not C." x exists iff C is irredundant.
- **Primal as optimization version** C is $l(x, y \dots) \leq a$, optimize l over $C_1 \wedge \cdots \wedge C_n$ and compare to a.
- **Dual** "Find $\lambda_i \geq 0$ such that $C = \sum_i \lambda_i C_i$."
  λ exist iff C is redundant.

If done for each of n constraints, quite costly.

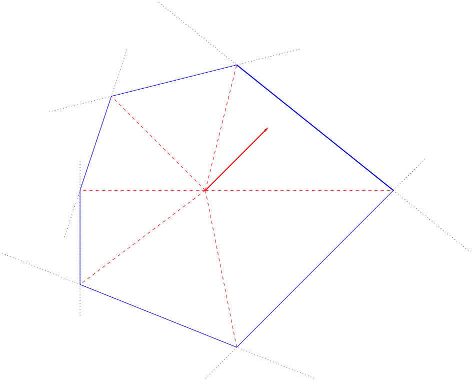
Plan

Ray-tracing, fast redundancy elimination



Maréchal & Périn (2017)

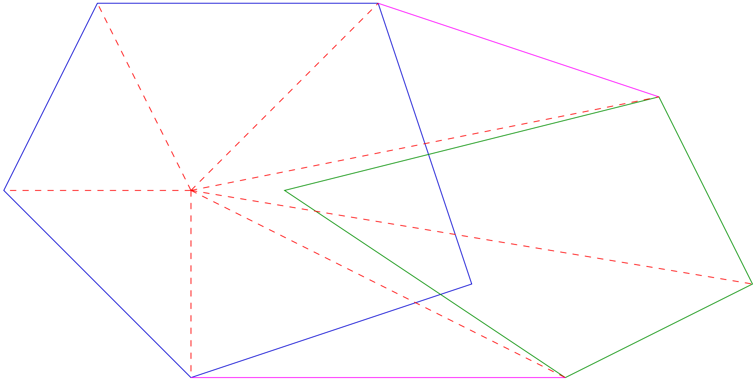
Parametric linear programming for projection



Parameters appearing linearly in the objective function: line of sight to face

Maréchal, 2017

Parametric linear programming for convex hull



Parameters appearing linearly in the objective function: line of sight to face

Maréchal, 2017

Fast parametric linear programming

- Parallel exploration of the region graph
- Use of floating-point for exact solving
- Elimination of redundant constraints of region using ray-tracing

Floating-point for exact solving

The simplex algorithm does not simply give a numeric solution!

It gives a **vertex** as the intersection of n constraints.

- The vertex can be recomputed exactly and checked if a true solution or not.
- In the basis defined by the constraints, the objective function should be "trivially" at an optimum (all coefficients negative / positive). This can be computed exactly.

Our solution

- Call off-the-shelf floating-point linear programming solver (exploration in floating-point)
- Reconstruct in exact precision (linear arithmetic $Ax = b$) the vertex and optimality witness.

Gratuitous advertisement

https://github.com/VERIMAG-Polyhedra/VPL

Alexis Fouilhé · Alexandre Maréchal
Sylvain Boulmé · David Monniaux · Michaël Périn · Hang Yu

Plan

Are heuristics truly necessary?

Input

A control flow graph with arithmetic transitions
A bad control state

Output

Yes / no
Can we decorate the control-flow graph inductive polyhedral invariants with $\emptyset$ on the bad state?

Warning

This is **not** the same as "is the bad state reachable?", clearly undecidable.

One control state suffices

Idea: encode control state $1 \leq i \leq n$ as a vertex of a simplex with n vertices over extra variables.
Use guards "am I on this vertex? then do..."

"Convexification" add points in the middle, weeded out by the guards.

This simulates the original problem exactly.

Monniaux, 2018

Some undecidability

Undecidable if polynomial guards are allowed.

Idea:

- Add two variables i, j: each step $i+ = i + 1$, $j+ = j + i$, so (i, j) draw a parabola.
- All points added by the “convexification” lie above the parabola $j = P(i)$.
- Conjoin guards $j \leq P(i)$ to remove these spurious points.
- A convex polyhedral invariant exists if and only if the program terminates.
- Invariant = convex hull of reachable points.

Open question if only linear transitions.

Monniaux, 2018

Questions?

Advertisement: need a Coq developer for a CompCert backend for a secure processor