NLVerify

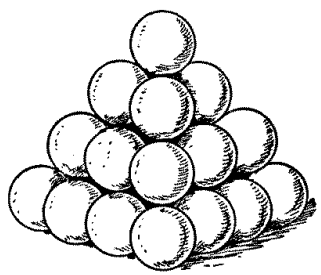
Verification of Polynomial Inequalities using Formal Floating-point Arithmetic

Victor Magron (CNRS VERIMAG)

Joint work with Tillmann Weisser and Benjamin Werner

INRIA Spades Seminar

October 27, 2015



Question:

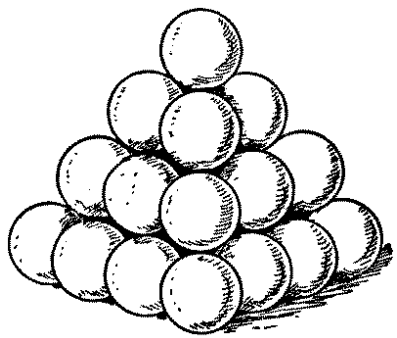
Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Applications:

- Formal proofs, e. g. Hales proof of the Kepler Conjecture
  - Completed in 2014
  - ~1000 non linear inequalities
  - took ~5000 CPU hours



- Software verification
- System control

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure

Naive Interval Enclosure:

$$y - 2x + 1 \in [0,1] - [2,2][0,1] + [1,1]$$

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure

Naive Interval Enclosure:

$$y - 2x + 1 \in [0,1] - [2,2][0,1] + [1,1] \subseteq [-1,2]$$

(very coarse, very fast, does not use Hypothesis)

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure
- Sums-of-Squares
  - Micromega (Besson)
  - NLCertify (Magron)
  - NLVerify (this work)

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure
- Sums-of-Squares
  - Micromega (Besson)
  - NLCertify (Magron)
  - NLVerify (this work)

Sums-of-Squares:

$$y - 2x + 1 = (x - 1)^2 + (y - x^2)$$

Automated formal proofs for questions like:

Does $0 \leq x, y \leq 1 \wedge x^2 \leq y$ imply $y - 2x + 1 \geq 0$?

Existing Methods:

- Taylor/Interval Method
  - Solovyev (HOL)
  - Melquiond (COQ)
- Bernstein Polynomials
- Quantifier Elimination
- Naive Interval Enclosure
- Sums-of-Squares
  - Micromega (Besson)
  - NLCertify (Magron)
  - NLVerify (this work)

Sums-of-Squares:

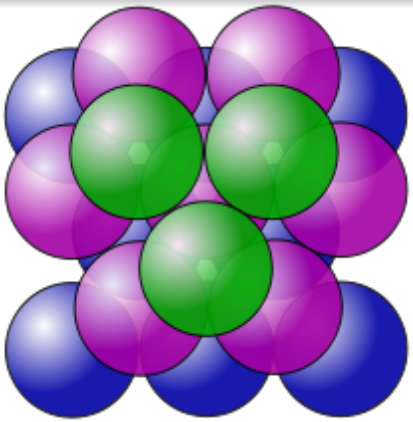
$$y - 2x + 1 = (x - 1)^2 + (y - x^2)$$

tighter but slower, certificate can be computed by external oracles

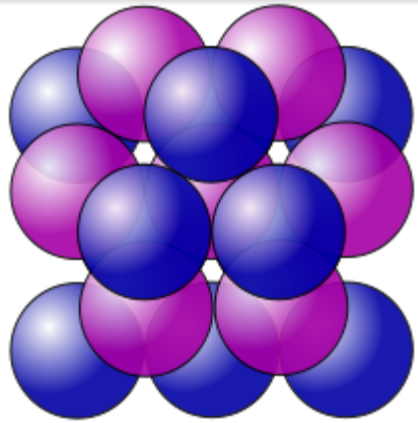
From Oranges Stack...

Kepler Conjecture (1611):

The maximal density of sphere packings in 3D-space is $\frac{\pi}{\sqrt{18}}$



Face-centered cubic Packing



Hexagonal Compact Packing

...to Flyspeck Nonlinear Inequalities

- The proof of T. Hales (1998) contains mathematical and computational parts
- Computation: check thousands of nonlinear inequalities
- **Flyspeck** [Hales 06]: **F**ormal **P**roof of **K**epler Conjecture

...to Flyspeck Nonlinear Inequalities

- The proof of T. Hales (1998) contains mathematical and computational parts
- Computation: check thousands of nonlinear inequalities
- **Flyspeck** [Hales 06]: **F**ormal **P**roof of **K**epler Conjecture
- **Project Completion on August 2014 by the Flyspeck team**

A "Simple" Example

In the computational part:

- Multivariate Polynomials:

  $\Delta\mathbf{x} :=$

  $x_1x_4(-x_1+x_2+x_3-x_4+x_5+x_6)+x_2x_5(x_1-x_2+x_3+x_4-x_5+x_6)+$
  $x_3x_6(x_1+x_2-x_3+x_4+x_5-x_6)-x_2(x_3x_4+x_1x_6)-x_5(x_1x_3+x_4x_6)$

A "Simple" Example

In the computational part:

- Semialgebraic functions: composition of polynomials with $|\cdot|, \sqrt{}, +, -, \times, /, \sup, \inf, \ldots$

$$p(\mathbf{x}) := \partial_4 \Delta \mathbf{x} \qquad q(\mathbf{x}) := 4x_1 \Delta \mathbf{x} \qquad r(\mathbf{x}) := p(\mathbf{x}) / \sqrt{q(\mathbf{x})}$$

$$l(\mathbf{x}) := -\frac{\pi}{2} + 1.6294 - 0.2213\,(\sqrt{x_2} + \sqrt{x_3} + \sqrt{x_5} + \sqrt{x_6} - 8.0) + 0.913\,(\sqrt{x_4} - 2.52) + 0.728\,(\sqrt{x_1} - 2.0)$$

A "Simple" Example

In the computational part:

- Transcendental functions $\mathcal{T}$: composition of semialgebraic functions with arctan, exp, sin, $+, -, \times, \ldots$

A "Simple" Example

In the computational part:

- Feasible set $K_{\text{box}} := [4, 6.3504]^3 \times [6.3504, 8] \times [4, 6.3504]^2$

Lemma$_{9922699028}$ from Flyspeck:

$$\forall \mathbf{x} \in K_{\text{box}}, \arctan\left(\frac{p(\mathbf{x})}{\sqrt{q(\mathbf{x})}}\right) + l(\mathbf{x}) \geq 0$$

New Framework (in my PhD thesis)

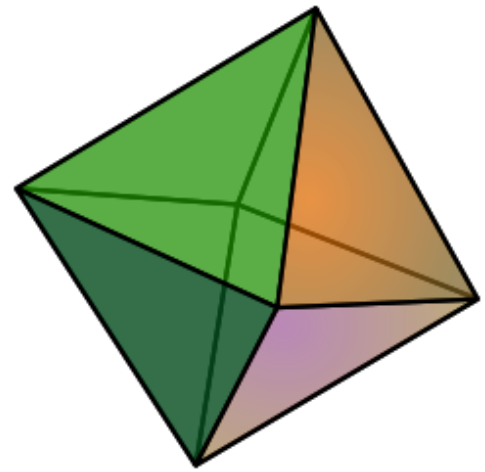
- Certificates for Nonlinear Optimization using SDP and:
  - Maxplus approximation (Optimal Control)
  - Nonlinear templates (Static Analysis)
- Verification of these certificates inside *Coq*:
  $p = \sigma_0 + \sum_j \sigma_j g_j \implies \forall \mathbf{x} \in K_{\text{box}}, \quad p(\mathbf{x}) \geq 0\,.$

A hint on what the oracle is doing:

- Linear Programming (LP):

$$\begin{aligned} \inf_{z} \quad & c^\top z \\ \text{s.t.} \quad & F z \geq 0 \ . \end{aligned}$$

  - Linear cost c
  - Linear inequalities "$\sum_i F_{ij} z_j \geq 0$"



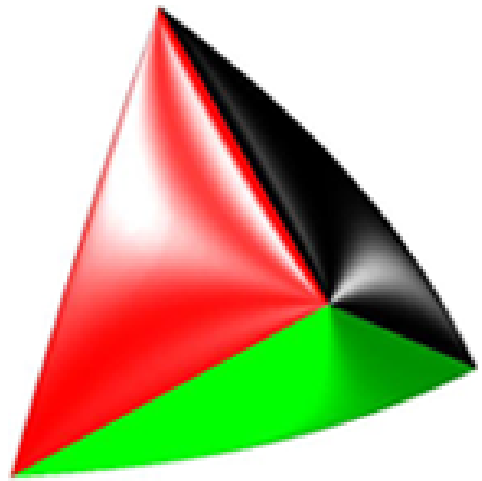
Polyhedron

A hint on what the oracle is doing:

- Semidefinite Programming (SDP):

$$\begin{aligned} \inf_{z} \quad & c^\top z \\ \text{s.t.} \quad & \sum_i F_i z_i \succeq 0 \ . \end{aligned}$$

  - Linear cost c
  - Symmetric matrices F_i
  - Linear matrix inequalities "$F \succeq 0$" (F has nonnegative eigenvalues)



Spectrahedron

A hint on what the oracle is doing:

Finding an SOS-representation boils down to solving an SDP.

A hint on what the oracle is doing:

Finding an SOS-representation boils down to solving an SDP.

There are efficient solvers available to solve SDPs.
(SDPA, CSDP, SDPT3, SeDuMi, Mosek,...)

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X & Y \end{pmatrix} \begin{pmatrix} z_1 & z_2 & z_4 \\ z_2 & z_3 & z_5 \\ z_4 & z_5 & z_6 \end{pmatrix} \begin{pmatrix} 1 \\ X \\ Y \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X & Y \end{pmatrix} \begin{pmatrix} z_1 & z_2 & z_4 \\ z_2 & z_3 & z_5 \\ z_4 & z_5 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ X \\ Y \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X & Y \end{pmatrix} \begin{pmatrix} z_1 & z_2 & z_4 \\ z_2 & z_3 & 0 \\ z_4 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ X \\ Y \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X & Y \end{pmatrix} \begin{pmatrix} z_1 & z_2 & 0 \\ z_2 & z_3 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ X \\ Y \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & z_2 \\ z_2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & z_2 \\ z_2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{11} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_3 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- We restrict ourselves to representations of degree ≤ 2. Write

$$
\begin{aligned}
Y - 2X + 1 = & \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_7 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot X \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_8 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - X) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_9 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot Y \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (1 - Y) \\
& + \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} z_{10} - z_9 + 1 \end{pmatrix} \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)
\end{aligned}
$$

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that
$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that

$$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$$

- e. g. $z_7 = z_8 = z_9 = z_{10} = 0$.

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that

$$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$$

- e. g. $z_7 = z_8 = z_9 = z_{10} = 0$.
- Substituting the solution: $Y - 2X + 1 =$

$$\begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} + \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that

$$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$$

- e. g. $z_7 = z_8 = z_9 = z_{10} = 0$.
- Substituting the solution: $Y - 2X + 1 =$

$$\begin{pmatrix} 1 & X \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} \right) \begin{pmatrix} 1 \\ X \end{pmatrix} + \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)$$

Example:

Find SOS-decomposition for $0 \leq x, y \leq 1 \wedge x^2 \leq y \Rightarrow y - 2x + 1 \geq 0$!

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that

$$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$$

- e. g. $z_7 = z_8 = z_9 = z_{10} = 0$.
- Substituting the solution: $Y - 2X + 1 =$

$$\left(\begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \right) + \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)$$

Example:

SOS-decomposition: $Y - 2X + 1 = (1 - X)^2 + (Y - X^2)$

- Find $z_7, z_8, z_9, z_{10} \geq 0$ such that

$$\begin{pmatrix} 1 - z_8 - z_{10} & (z_8 - z_7 - 2)/2 \\ (z_8 - z_7 - 2)/2 & z_{10} - z_9 + 1 \end{pmatrix} \succeq 0.$$

- e. g. $z_7 = z_8 = z_9 = z_{10} = 0$.
- Substituting the solution: $Y - 2X + 1 =$

$$\left(\begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ X \end{pmatrix} \right) + \begin{pmatrix} 1 \end{pmatrix} \cdot (Y - X^2)$$

Example:

SOS-decomposition: $Y - 2X + 1 = (1 - X)^2 + (Y - X^2)$

sdp solvers only find approximate certificates:

$$\begin{aligned} Y - 2X + 1 \quad \simeq \quad & 2.00014(0.707263X + 0.000078Y - 0.70695)^2 \\ + \quad & 0.000332(-0.408035X + 0.816664Y - 0.408126)^2 \\ + \quad & 0.000284Y + 0.000116(1 - Y) + 1.00034(Y - X^2) \end{aligned}$$

Example:

SOS-decomposition: $Y - 2X + 1 = (1 - X)^2 + (Y - X^2)$

sdp solvers only find approximate certificates:

$$\begin{aligned} Y - 2X + 1 \quad &\simeq \quad 2.00014(0.707263X + 0.000078Y - 0.70695)^2 \\ &+ \quad 0.000332(-0.408035X + 0.816664Y - 0.408126)^2 \\ &+ \quad 0.000284Y + 0.000116(1 - Y) + 1.00034(Y - X^2) \end{aligned}$$

Exact error polynomial

$$\begin{aligned} \varepsilon \quad &:= \quad 0.000232209X^2 - 5.81334 \times 10^{-7}XY - 0.0000297356X \\ &+ \quad 0.000221436Y^2 + 0.0000621035Y - 0.000201126 \end{aligned}$$

Example:

SOS-decomposition: $Y - 2X + 1 = (1 - X)^2 + (Y - X^2)$

sdp solvers only find approximate certificates:

$$\begin{aligned} Y - 2X + 1 \quad &\simeq \quad 2.00014(0.707263X + 0.000078Y - 0.70695)^2 \\ &+ \quad 0.000332(-0.408035X + 0.816664Y - 0.408126)^2 \\ &+ \quad 0.000284Y + 0.000116(1 - Y) + 1.00034(Y - X^2) \end{aligned}$$

Exact error polynomial

$$\begin{aligned} \varepsilon \quad &:= \quad 0.000232209X^2 - 5.81334 \times 10^{-7}XY - 0.0000297356X \\ &+ \quad 0.000221436Y^2 + 0.0000621035Y - 0.000201126 \end{aligned}$$

How can we employ such numerical certificates for formal verification?

How to use numerical certificates in Coq?

tactic	strategy

$$\begin{aligned}\varepsilon \quad &:= \quad 0.000232209X^2 - 5.81334 \times 10^{-7}XY - 0.0000297356X \\ &+ \quad 0.000221436Y^2 + 0.0000621035Y - 0.000201126\end{aligned}$$

How to use numerical certificates in Coq?

tactic	strategy
Micromega	uses heuristics to get an exact representation

$$\varepsilon \quad = \quad 0$$

How to use numerical certificates in Coq?

tactic	strategy
Micromega	uses heuristics to get an exact representation
NLCertify	gives lower bound on ε by exact computations

$$\begin{aligned} \varepsilon^* \quad := \quad & 0.000232209X^2 - 5.81334 \times 10^{-7} XY - 0.0000297356X \\ + \quad & 0.000221436Y^2 + 0.0000621035Y - 0.000201126 \end{aligned}$$

How to use numerical certificates in Coq?

tactic	strategy
Micromega	uses heuristics to get an exact representation
NLCertify	gives lower bound on ε by exact computations
NLVerify	use interval arithmetics to bound ε

$$\varepsilon^* \quad := \quad \text{enclosure of } \varepsilon$$

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\}.$$

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \subseteq K_{\text{box}},$$

where $K_{\text{box}} = [\mathbf{a}, \mathbf{b}]$, with $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^n$ (plus assumption on the g_j).

We are interested in the fact of f being *non negative* on K_{pop}, i. e.

$$\forall \mathbf{x} \in K_{\text{pop}} : f(\mathbf{x}) \geq 0.$$

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \subseteq K_{\text{box}},$$

where $K_{\text{box}} = [\mathbf{a}, \mathbf{b}]$, with $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^n$ (plus assumption on the g_j).

We are interested in the fact of f being *non negative* on K_{pop}, i. e.

$$\forall \mathbf{x} \in K_{\text{pop}} : f(\mathbf{x}) \geq 0.$$

Number formats

$\mathbb{R}$	axiomatic

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \subseteq K_{\text{box}},$$

where $K_{\text{box}} = [\mathbf{a}, \mathbf{b}]$, with $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^n$ (plus assumption on the g_j).

We are interested in the fact of f being *non negative* on K_{pop}, i. e.

$$\forall \mathbf{x} \in K_{\text{pop}} : f(\mathbf{x}) \geq 0.$$

Number formats

$\mathbb{R}$	axiomatic
$\mathbb{Q}$	exact, slow

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \subseteq K_{\text{box}},$$

where $K_{\text{box}} = [\mathbf{a}, \mathbf{b}]$, with $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^n$ (plus assumption on the g_j).

We are interested in the fact of f being *non negative* on K_{pop}, i. e.

$$\forall \mathbf{x} \in K_{\text{pop}} : f(\mathbf{x}) \geq 0.$$

Number formats

$\mathbb{R}$	axiomatic
$\mathbb{Q}$	exact, slow
$\mathbb{F}$	fast, certified inside Coq (Flocq, Boldo/Melquiond), rounding errors

General Framework

Consider n-variate polynomials $f, g_0, \ldots, g_m \in \mathbb{Q}[\mathbf{X}]$ and a compact set

$$K_{\text{pop}} := \{\mathbf{x} \in \mathbb{R}^n \mid g_0(\mathbf{x}) \geq 0, \ldots, g_m(\mathbf{x}) \geq 0\} \subseteq K_{\text{box}},$$

where $K_{\text{box}} = [\mathbf{a}, \mathbf{b}]$, with $\mathbf{a}, \mathbf{b} \in \mathbb{Q}^n$ (plus assumption on the g_j).

We are interested in the fact of f being *non negative* on K_{pop}, i. e.

$$\forall \mathbf{x} \in K_{\text{pop}} : f(\mathbf{x}) \geq 0.$$

Number formats

$\mathbb{R}$	axiomatic
$\mathbb{Q}$	exact, slow
$\mathbb{F}$	fast, certified inside Coq (Flocq, Boldo/Melquiond), rounding errors
$\mathbb{I}$	to keep track of rounding errors

Notations

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p.

Notations

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p.
  We are using one precision for all operations.
  In this talk $r = 10$, in the implementation $r = 2$.

Notations

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p.
  We are using one precision for all operations.
  In this talk $r = 10$, in the implementation $r = 2$.
- Intervals $\mathbb{I}_p := \mathbb{I}_{r,p}$ with floating point bounds $\mathbb{F}_p$.

Notations

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p.
  We are using one precision for all operations.
  In this talk $r = 10$, in the implementation $r = 2$.
- Intervals $\mathbb{I}_p := \mathbb{I}_{r,p}$ with floating point bounds $\mathbb{F}_p$.

We map rationals to intervals via the *enclosure*

$$\mathbb{Q} \to \mathbb{I}_p,\ [a]_p := \left[\max_{x \in \mathbb{F}_p} \{x \mid x \leq a\}\ ,\ \min_{x \in \mathbb{F}_p} \{x \mid x \geq a\} \right].$$

Notations

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p.
  We are using one precision for all operations.
  In this talk $r = 10$, in the implementation $r = 2$.
- Intervals $\mathbb{I}_p := \mathbb{I}_{r,p}$ with floating point bounds $\mathbb{F}_p$.

We map rationals to intervals via the *enclosure*

$$\mathbb{Q} \to \mathbb{I}_p,\ [a]_p := \left[\max_{x \in \mathbb{F}_p} \{x \mid x \leq a\}\ ,\ \min_{x \in \mathbb{F}_p} \{x \mid x \geq a\} \right].$$

Attention!

Interval arithmetic does not carry any ring structure. The enclosure does not commute with the operations in $\mathbb{Q}$. In general:

$$[a + b]_p \subsetneq [a]_p + [b]_p\,.$$

Two applications of intervals on polynomials

- Replace coefficients by intervals to speed up computation.
- Replace variables by intervals to obtain bounds on the function.

Two applications of intervals on polynomials

- Replace coefficients by intervals to speed up computation.
- Replace variables by intervals to obtain bounds on the function.

Two applications of intervals on polynomials

- Replace coefficients by intervals to speed up computation.

Coefficient Enclosure

Building a *coefficient enclosure* of a polynomial $f \in \mathbb{Q}[\mathbf{X}]$ is done by mapping its coefficients to the corresponding intervals via $[\bullet]_p$. If $f = \sum_\alpha f_\alpha \mathbf{X}^\alpha$, its coefficient enclosure is the set of polynomials

$$[f]_p = \sum_\alpha [f_\alpha]_p \, \mathbf{X}^\alpha := \left\{ \sum_\alpha \hat{f}_\alpha \mathbf{X}^\alpha \mid \hat{f}_\alpha \in [f_\alpha]_p \right\}$$

Two applications of intervals on polynomials

- Replace coefficients by intervals to speed up computation.

Coefficient Enclosure

Building a *coefficient enclosure* of a polynomial $f \in \mathbb{Q}[\mathbf{X}]$ is done by mapping its coefficients to the corresponding intervals via $[\bullet]_p$. If $f = \sum_\alpha f_\alpha \mathbf{X}^\alpha$, its coefficient enclosure is the set of polynomials

$$[f]_p = \sum_\alpha [f_\alpha]_p \, \mathbf{X}^\alpha := \left\{ \sum_\alpha \hat{f}_\alpha \mathbf{X}^\alpha \mid \hat{f}_\alpha \in [f_\alpha]_p \right\}$$

Keep in mind!

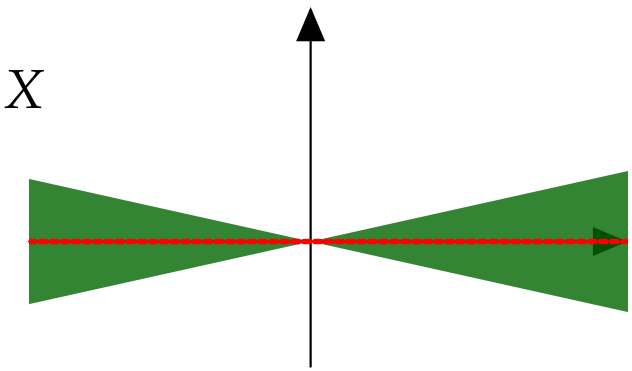
The coefficient enclosure depends on the representation of f.

Two applications of intervals on polynomials

- Replace coefficients by intervals to speed up computation.

Coefficient Enclosure

$$
\begin{aligned}
f &:= \frac{1}{3}X - \frac{1}{3}X = 0 \\
[f]_2 &= [0.33, 0.34]X - [0.33, 0.34]X \\
[0]_2 &= [0, 0]
\end{aligned}
$$



Two applications of intervals on polynomials

- Replace variables by intervals to obtain bounds on the function.

Variable Enclosure

The *variable enclosure* $|f|_{\mathbf{B}}$ of a polynomial f with respect to a hyper box $\mathbf{B} = (I_1 \cdots, I_n) \subseteq \mathbb{I}_p^n$ is built by replacing every variable X_i by the corresponding interval I_i. If $f = \sum_\alpha f_\alpha \mathbf{X}^\alpha$, its variable enclosure is

$$|f|_{\mathbf{B}} = \sum_\alpha f_\alpha \mathbf{B}^\alpha \subseteq \mathbb{I}_p$$

Two applications of intervals on polynomials

- Replace variables by intervals to obtain bounds on the function.

Variable Enclosure

The *variable enclosure* $|f|_{\mathbf{B}}$ of a polynomial f with respect to a hyper box $\mathbf{B} = (I_1 \cdots, I_n) \subseteq \mathbb{I}_p^n$ is built by replacing every variable X_i by the corresponding interval I_i. If $f = \sum_\alpha f_\alpha \mathbf{X}^\alpha$, its variable enclosure is

$$|f|_{\mathbf{B}} = \sum_\alpha f_\alpha \mathbf{B}^\alpha \subseteq \mathbb{I}_p$$

Of course:

The variable enclosure depends on the representation of f.

Two applications of intervals on polynomials

- Replace variables by intervals to obtain bounds on the function.

Variable Enclosure

Let $\mathbf{B} = [-1,1] \times [0,1] \times [0,1]$. Then

$$\begin{aligned} X(Y-Z) &= XY - YZ \\ |X(Y-Z)|_{\mathbf{B}} &= [-1,1][-1,1] = [-1,1] \\ |XY - XZ|_{\mathbf{B}} &= [-1,1] - [-1,1] = [-2,2] \end{aligned}$$

Two applications of intervals on polynomials

- COEFFICIENT ENCLOSURE
- VARIABLE ENCLOSURE

We are combining both methods and SOS-certification.

Coq Implementation

Theorem:

$$\left|[f]_p\right|_{K_{\text{box}}} \subseteq [\ell, \infty) \Rightarrow f \geq \ell \text{ on } K_{\text{box}}.$$

Coq Implementation

Theorem:

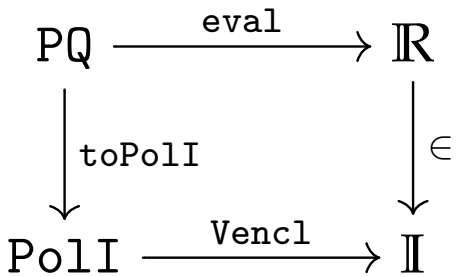
$$\left|[f]_p\right|_{K_{\text{box}}} \subseteq [\ell, \infty) \Rightarrow f \geq \ell \text{ on } K_{\text{box}}.$$

CoqVersion:

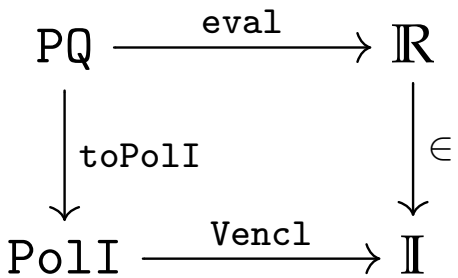
```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

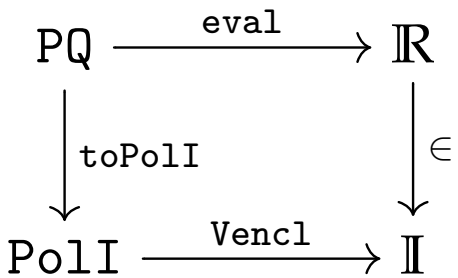


```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```




```
Inductive PQ :=
| PEc  : ℚ → PQ
| PEx  : positive → PQ
| PEadd: PQ -> PQ → PQ
| PEsub: PQ -> PQ → PQ
| ... .
```

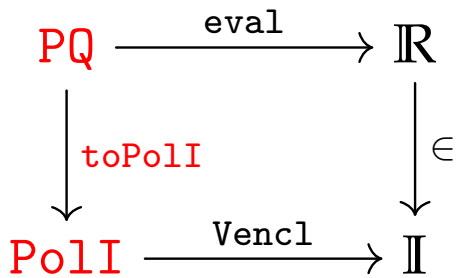
```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```




```
Inductive PQ :=
| PEc  : ℚ → PQ
| PEx  : positive → PQ
| PEadd: PQ -> PQ → PQ
| PEsub: PQ -> PQ → PQ
| ... .
```

```
Inductive PolI :=
| IPc  : 𝕀 → PolI
| IPinj: positive → PolI → PolI
| IPX  : PolI → positive → PolI → PolI.
```

```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

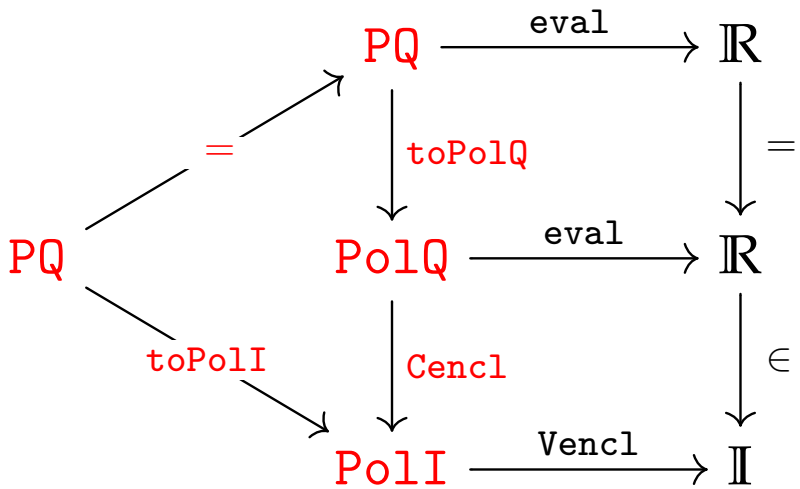



```
Inductive PQ :=
| PEc  : ℚ → PQ
| PEx  : positive → PQ
| PEadd: PQ -> PQ → PQ
| PEsub: PQ -> PQ → PQ
| ... .
```

```
Inductive PolI :=
| IPc  : 𝕀 → PolI
| IPinj: positive → PolI → PolI
| IPX  : PolI → positive → PolI → PolI.
```

```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

Proof:

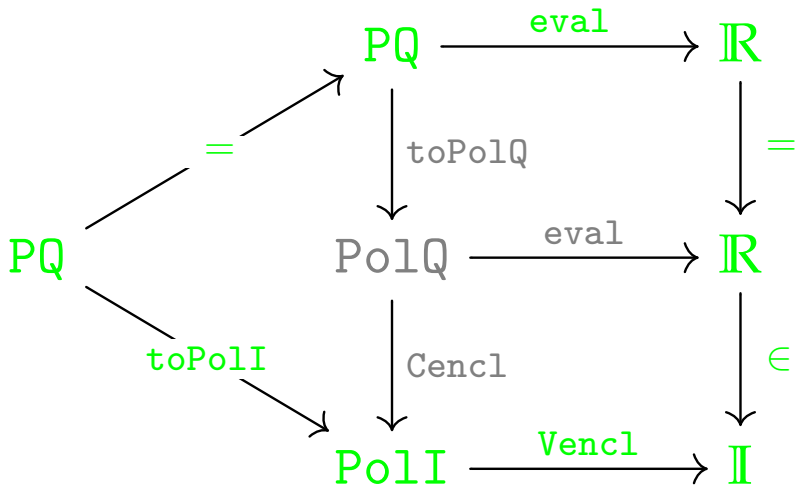



```
Inductive PQ :=
| PEc  : ℚ → PQ
| PEx  : positive → PQ
| PEadd: PQ -> PQ → PQ
| PEsub: PQ -> PQ → PQ
| ... .
```

```
Inductive PolQ :=
| QPc  : ℚ → PolQ
| QPinj: positive → PolQ → PolQ
| QPX  : PolQ → positive → PolQ → PolQ.
```

```
Lemma toPolI_ok p box pt :
pt ∈ box → eval pt p ∈ Vencl box (toPolI p).
```

NLVerify:




```
Inductive PQ :=
| PEc  : ℚ → PQ
| PEx  : positive → PQ
| PEadd: PQ -> PQ → PQ
| PEsub: PQ -> PQ → PQ
| ... .
```

```
Inductive PolQ :=
| QPc  : ℚ → PolQ
| QPinj: positive → PolQ → PolQ
| QPX  : PolQ → positive → PolQ → PolQ.
```

Correctness lemmas for `PolI`

For `PolQ` a correctness lemma looks like:

```
Lemma QPadd_ok (p q: PolQ) pt :
eval pt (p !++ q) = eval pt p + eval pt q.
```

Correctness lemmas for `PolI`

For `PolQ` a correctness lemma looks like:

```
Lemma QPadd_ok (p q: PolQ) pt :
eval pt (p !++ q) = eval pt p + eval pt q.
```

The direct translation of this lemma to `PolI` is false.

Correctness lemmas for `PolI`

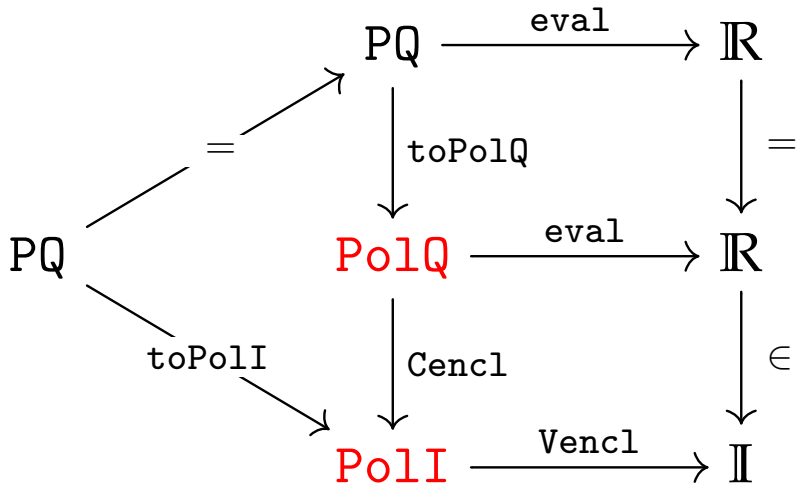
For `PolQ` a correctness lemma looks like:

```
Lemma QPadd_ok (p q: PolQ) pt :
eval pt (p !++ q) = eval pt p + eval pt q.
```

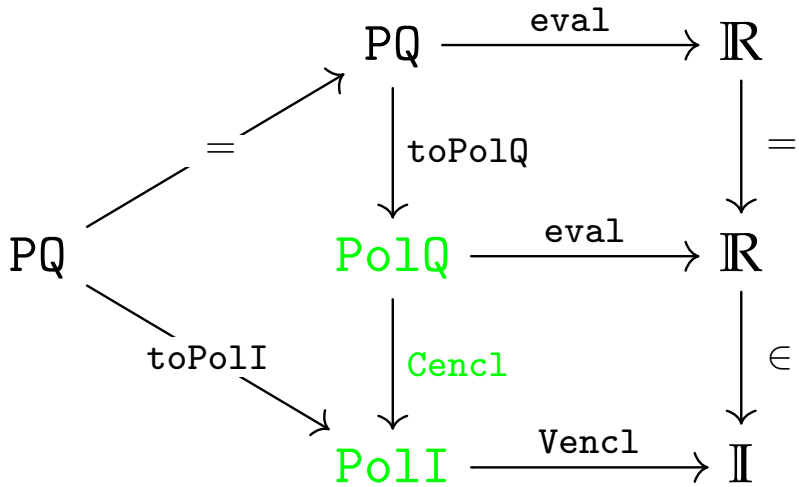
The direct translation of this lemma to `PolI` is false. A `PolI` operation can only be correct w.r.t. the underlying `PolQ` expressions:

```
Lemma Padd_coef_ok (p q: PolQ) (P Q: PolI) :
p ∈ P -> q ∈ Q -> (p !++ q) ∈ (P ?++ Q).
```

Correctness lemmas for `PolI`

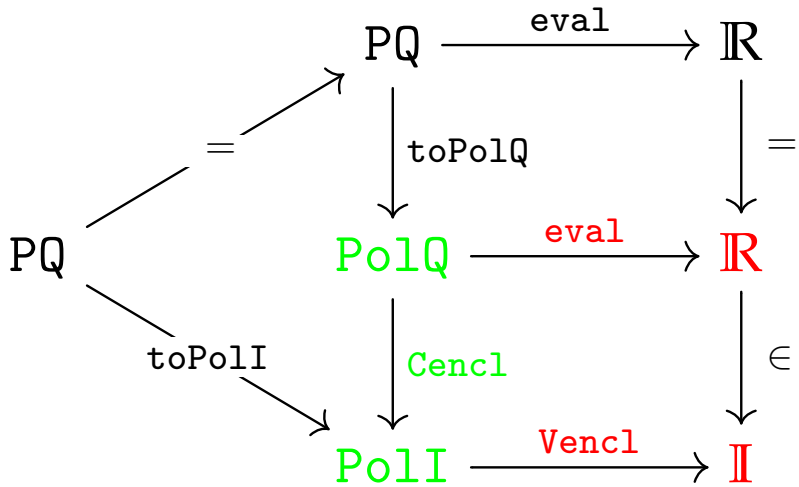


Correctness lemmas for PolI



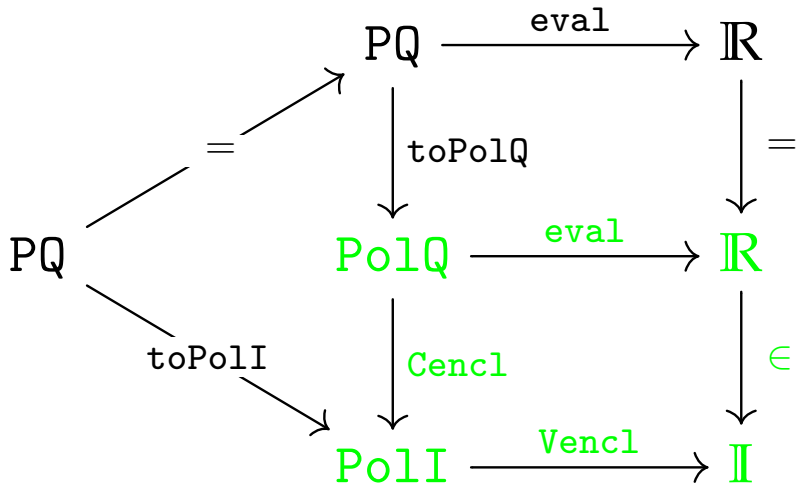
→ Correctness lemmas

Proof of Lemma



→ Correctness lemmas

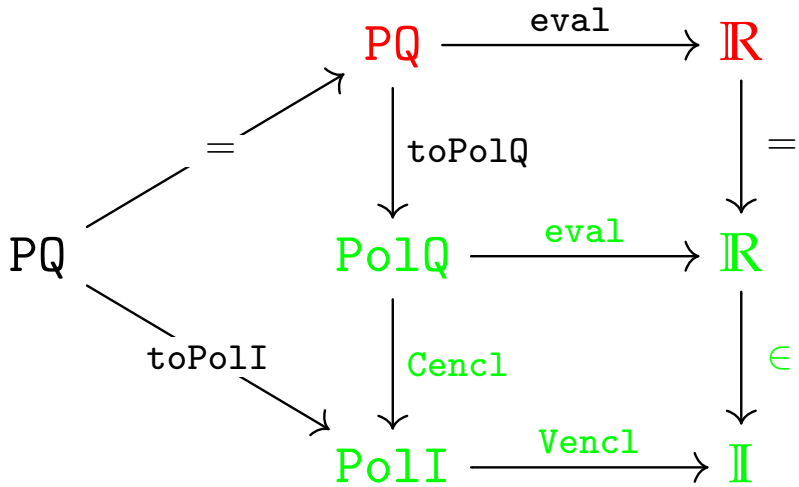
Proof of Lemma



→ Correctness lemmas

→ easy because of same structure

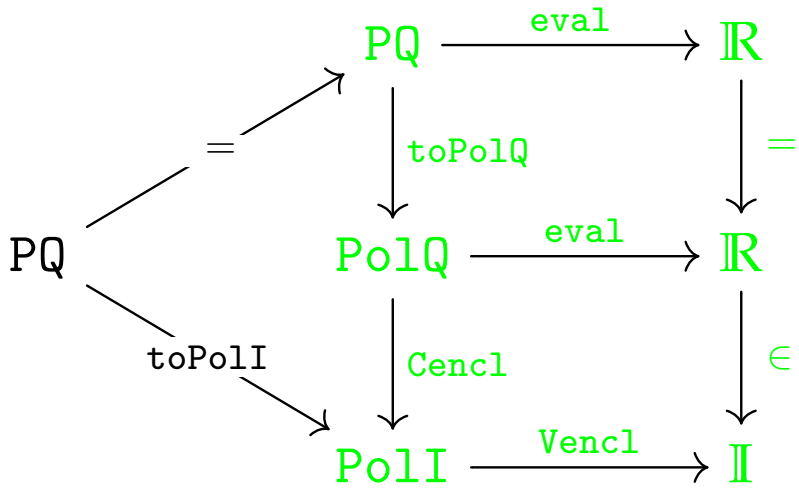
Proof of Lemma



→ Correctness lemmas

→ easy because of same structure

Proof of Lemma

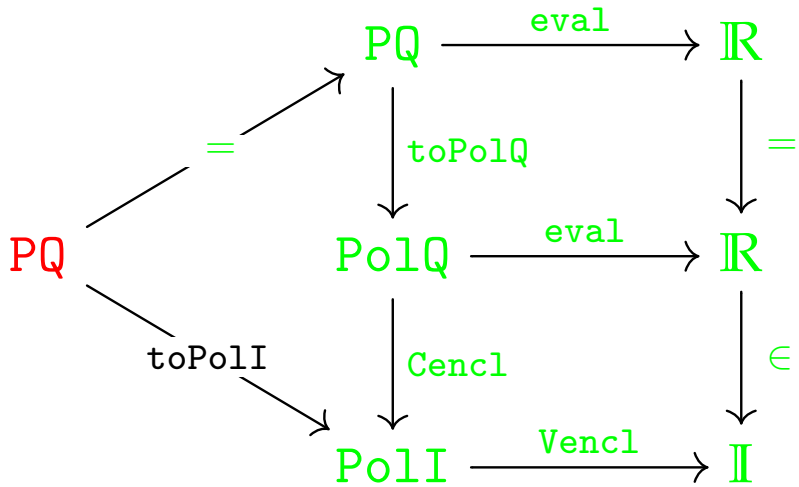


→ Correctness lemmas

→ easy because of same structure

→ basically proved in `ring`

Proof of Lemma

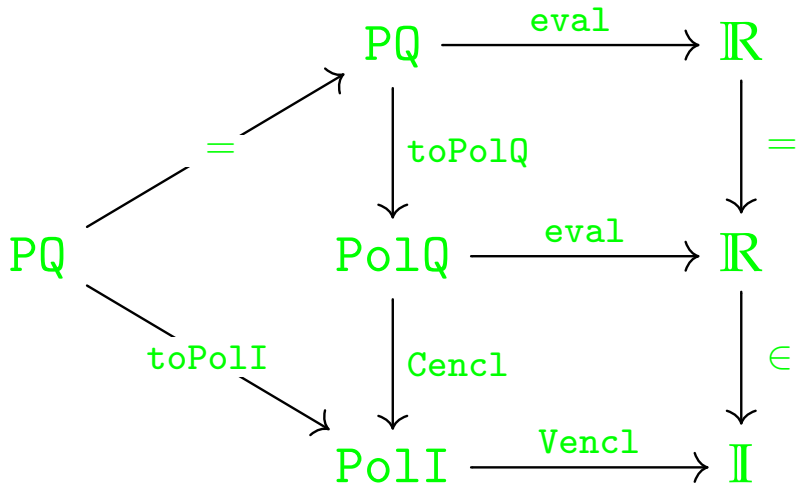


→ Correctness lemmas

→ easy because of same structure

→ basically proved in `ring`

Proof of Lemma



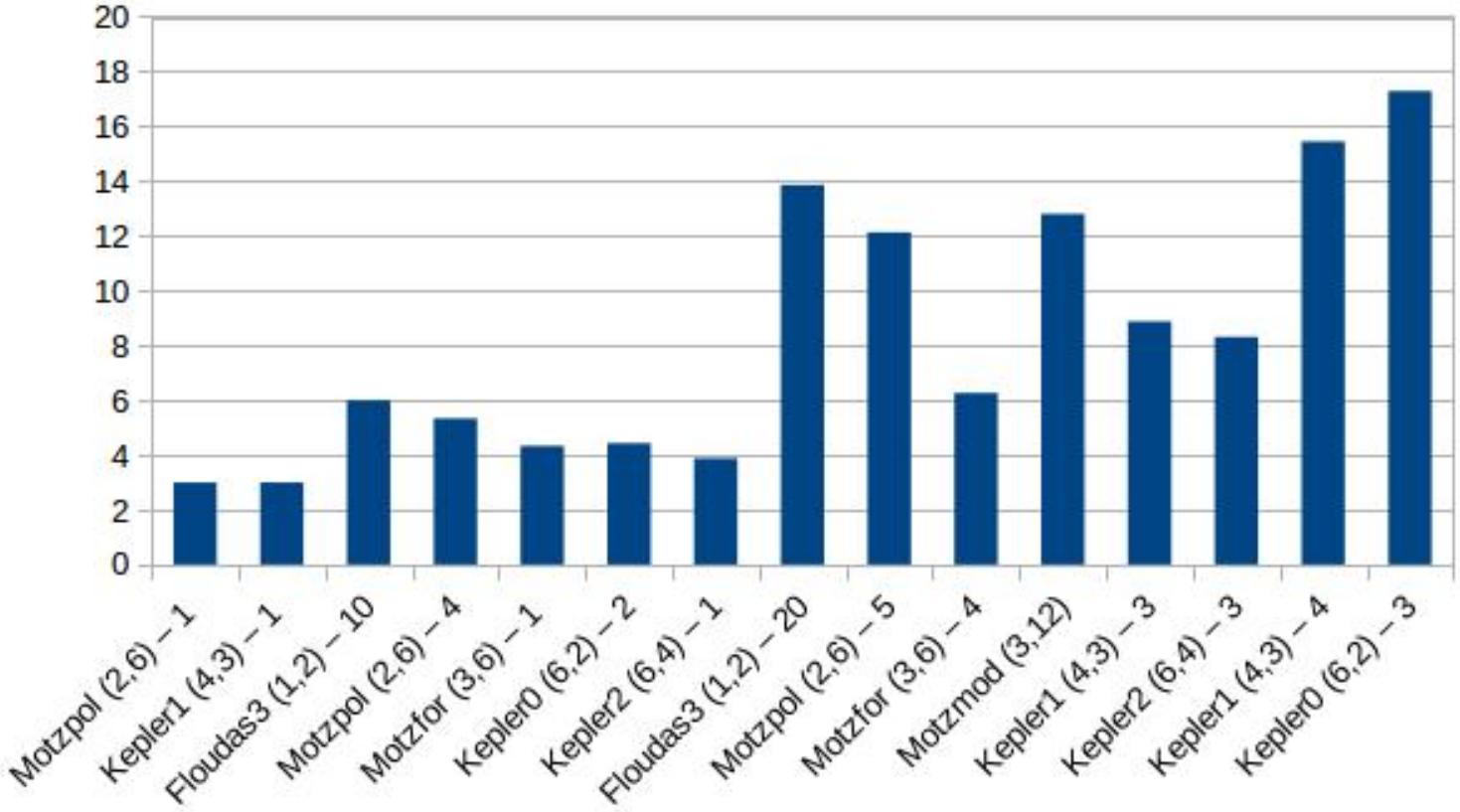
→ Correctness lemmas
→ easy because of same structure
→ basically proved in `ring`
→ follows from the correctness of interval arithmetic

Speedup NLVerify (p=50) vs. NLCertify

x-Axis: examples (ordered by time_nlc)
y-Axis: ratio time_nlv / time_nlc

Speedup NLVerify (p=50) vs. NLCertify

x-Axis: examples (ordered by time_nlc)
y-Axis: ratio time_nlv / time_nlc

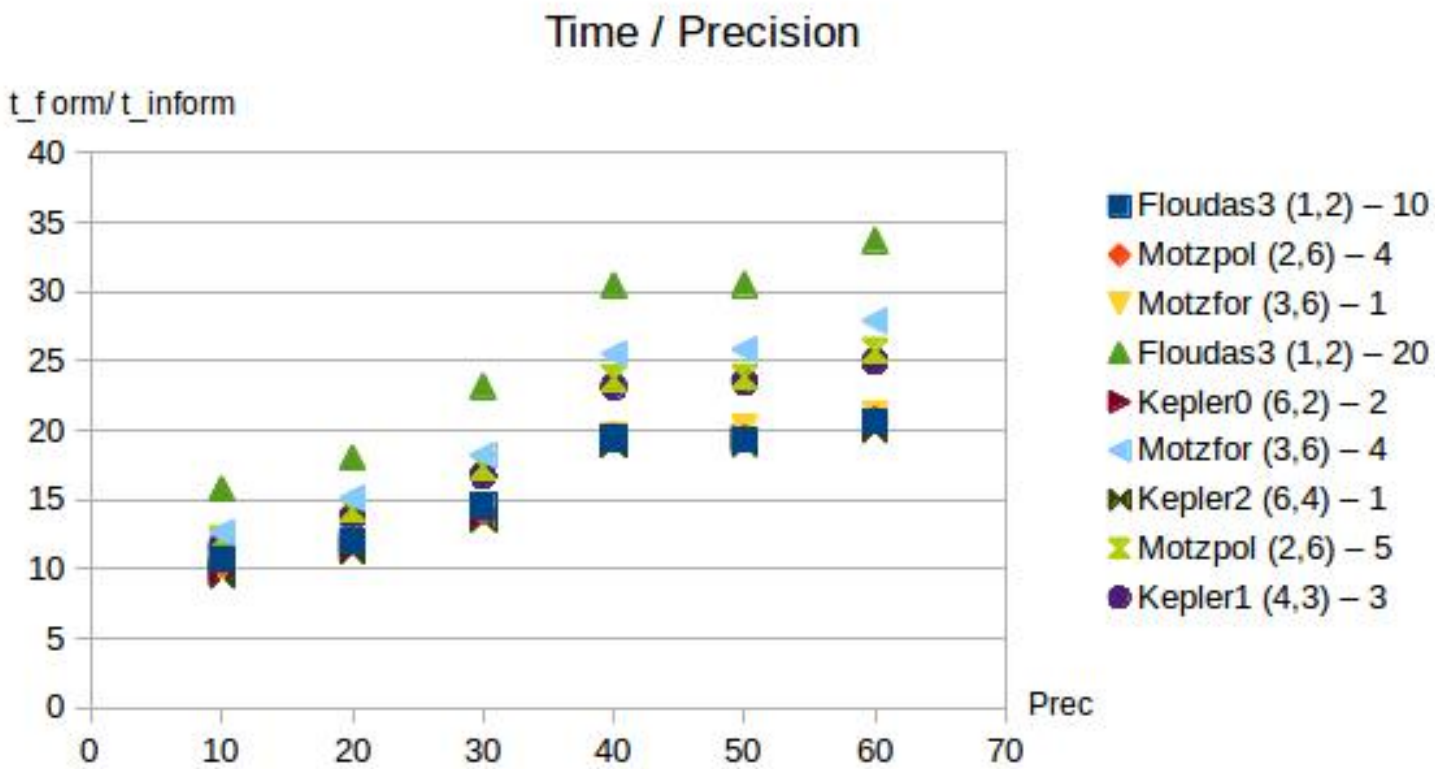


Speedup Decreasing Precision

x-Axis: precision
y-Axis: ratio time_formal / time_informal

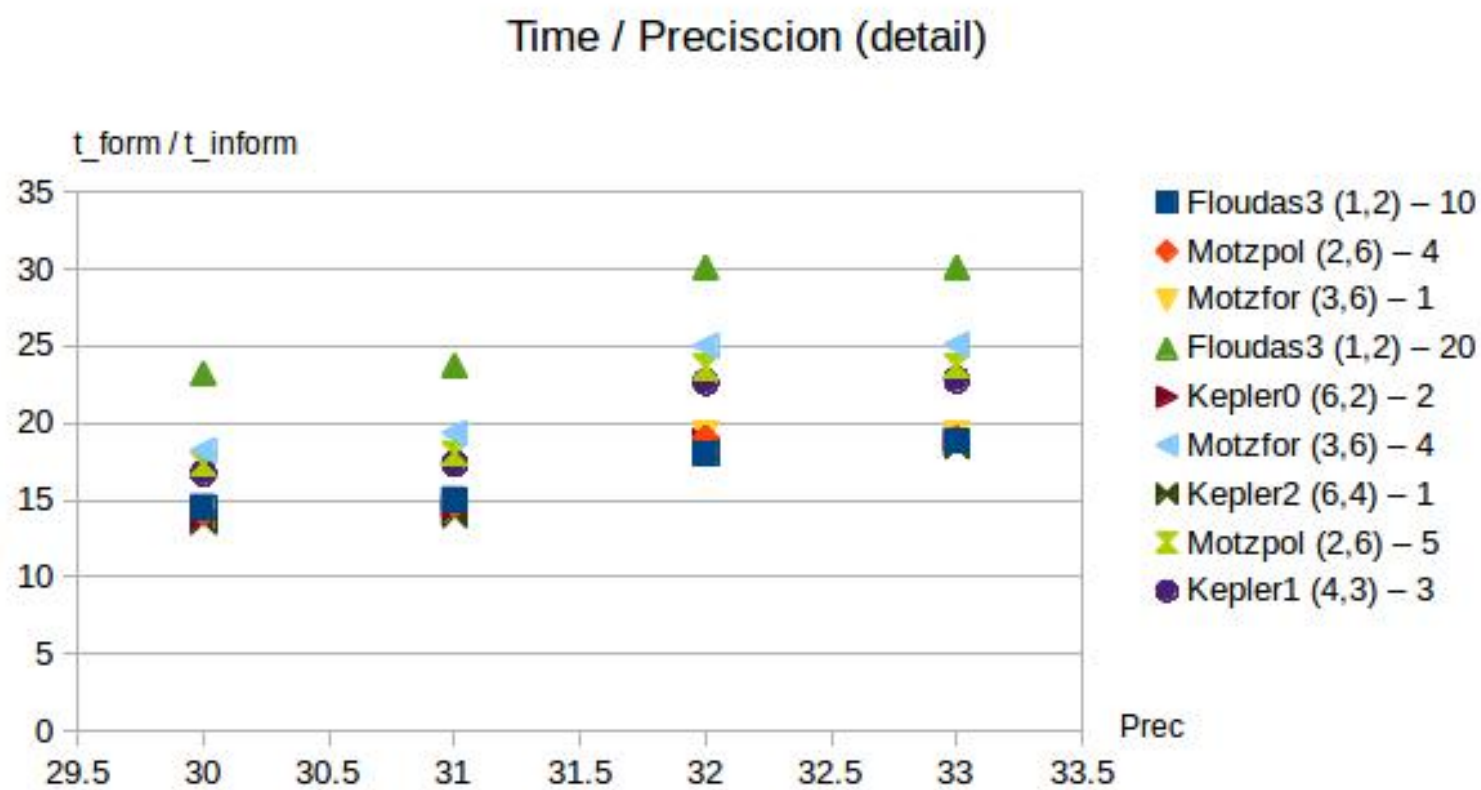
Speedup Decreasing Precision

x-Axis: precision
y-Axis: ratio time_formal / time_informal



Speedup Decreasing Precision (detail)

x-Axis: precision
y-Axis: ratio time_formal / time_informal



Precision vs. Accuracy w.r.t. NLCertify

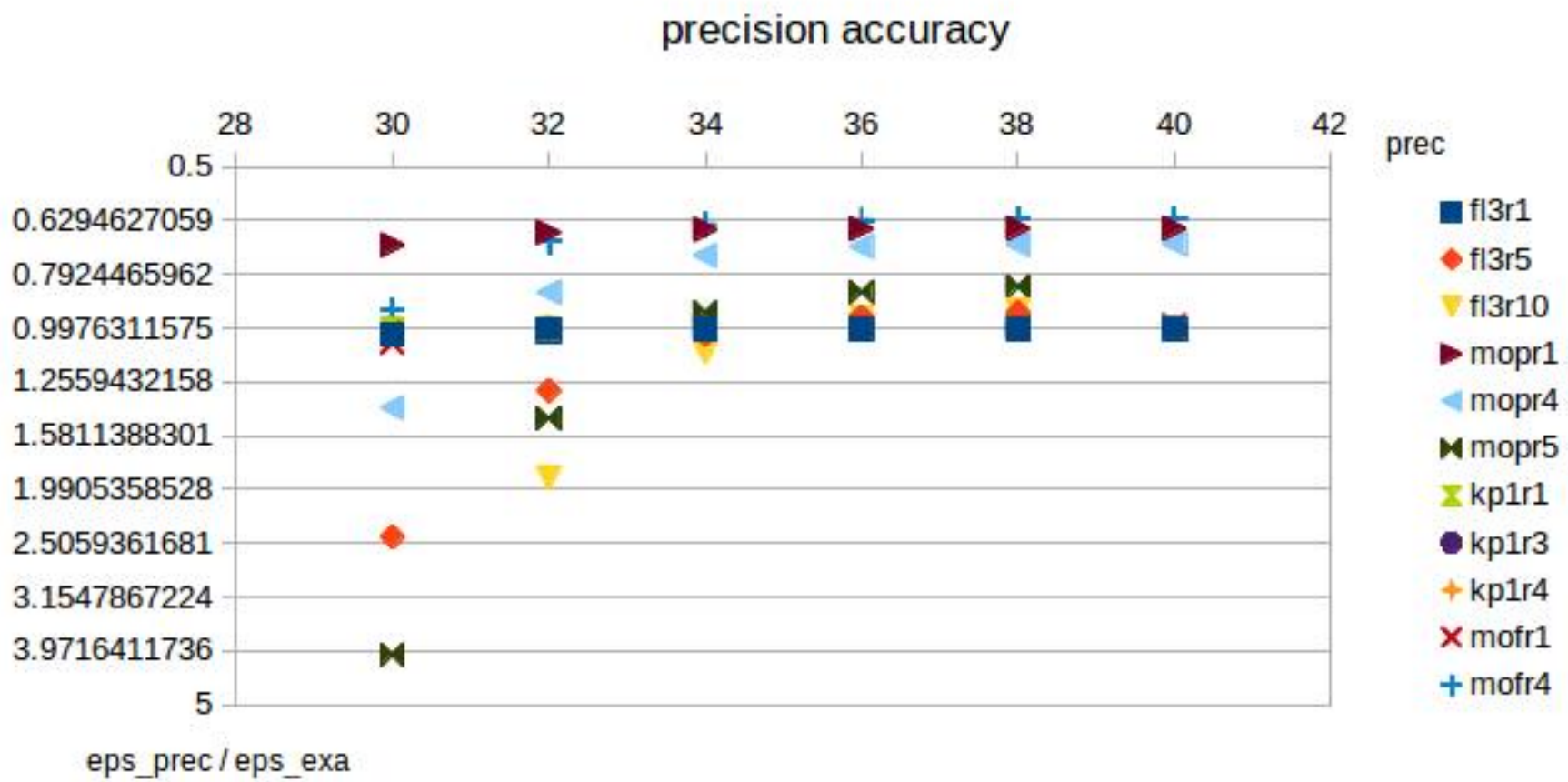
x-Axis: precision
y-Axis: ratio $\varepsilon_p / \varepsilon^*$

ε_p: loss with NLVerify precision p
ε^*: loss with NLCertify

Precision vs. Accuracy w.r.t. NLCertify

x-Axis: precision
y-Axis: ratio $\varepsilon_p / \varepsilon^*$

ε_p: loss with NLVerify precision p
ε^*: loss with NLCertify



Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Open: Can the performance be improved by using another SDP solver?

Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Open: Can the performance be improved by using another SDP solver?

Open: Is `PolI` the best polynomial representation for computation?

Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Open: Can the performance be improved by using another SDP solver?

Open: Is `PolI` the best polynomial representation for computation?

Open: Should we use different precisions for the different interval operations?

Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Open: Can the performance be improved by using another SDP solver?

Open: Is `PolI` the best polynomial representation for computation?

Open: Should we use different precisions for the different interval operations?

Open: Is there a better way to order arithmetic operations in the certificates?

Future Work

ToDo: Generalization to semi algebraic and transcendental functions

Open: Can the performance be improved by using another SDP solver?

Open: Is `PolI` the best polynomial representation for computation?

Open: Should we use different precisions for the different interval operations?

Open: Is there a better way to order arithmetic operations in the certificates?

Open: Would the computation be faster using 50bit words?

Primal-dual Moment-SOS in COQ

- $\mathcal{M}_+(\mathbf{S})$: space of probability measures supported on $\mathbf{S}$
- $\Sigma[\mathbf{x}]$: polynomial sums of squares

Theory:

(Primal)		(Dual)
$\inf \quad \int_{\mathbf{S}} f \, d\mu$	$=$	$\sup \quad \lambda$
s.t. $\quad \mu \in \mathcal{M}_+(\mathbf{S})$		s.t. $\quad \lambda \in \mathbb{R}$,
		$f - \lambda \in \Sigma[\mathbf{x}]$

Primal-dual Moment-SOS in Coq

- Finite moment sequences $\mathbf{z}$ of measures in $\mathcal{M}_+(\mathbf{S})$
- Truncated quadratic module $\Sigma_k[\mathbf{x}] := \Sigma[\mathbf{x}] \cap \mathbb{R}_{2k}[\mathbf{x}]$

Practice:

$$
\begin{array}{lll}
\text{(Moment)} & & \text{(SOS)} \\
\inf \quad \sum_\alpha f_\alpha \, \mathbf{z}_\alpha & = & \sup \quad \lambda \\
\text{s.t.} \quad \mathbf{M}_{k-v_j}(g_j \, \mathbf{z}) \succcurlyeq 0\,, \quad 0 \le j \le l, & & \text{s.t.} \quad \lambda \in \mathbb{R}\ , \\
\quad \mathbf{z}_1 = 1 & & \quad f - \lambda \in \Sigma_k[\mathbf{x}]
\end{array}
$$

End

Thank you for your attention!

http://www-verimag.imag.fr/~magron