LibPoly: A Library for Reasoning about Polynomials

Dejan Jovanović   Bruno Dutertre

SRI International

SMT Workshop 2017

Outline

Introduction

LibPoly

- Working with Polynomials
- Constructing a Sign Table
- Cylindrical Algebraic Decomposition

Conclusion

Outline

Introduction

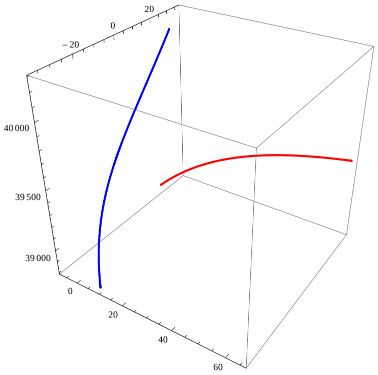
LibPoly

- Working with Polynomials
- Constructing a Sign Table
- Cylindrical Algebraic Decomposition

Conclusion

NON-LINEAR REASONING

MANY APPLICATIONS



$$T_1^x(t) = 3.2484 + 270.7t + 433.12t^2 - 324.83999t^3$$
$$T_1^y(t) = 15.1592 + 108.28t + 121.2736t^2 - 649.67999t^3$$
$$T_1^z(t) = 38980.8 + 5414t - 21656t^2 + 32484t^3$$

$$T_2^x(t) = 1.0828 - 135.35t + 234.9676t^2 2 + 3248.4t^3$$
$$T_2^y(t) = 18.40759 - 230.6364t - 121.2736t^2 - 649.67999t^3$$
$$T_2^z(t) = 40280.15999 - 10828t + 24061.9816t^2 - 32484t^3$$

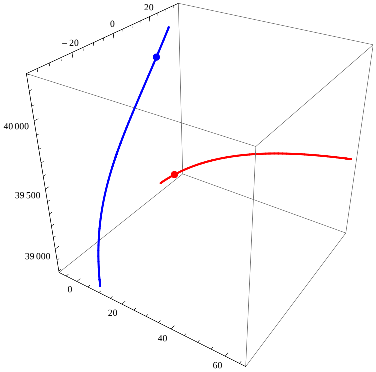
$$D = 5 \qquad H = 1000 \qquad 0 \leq t \leq \frac{1}{20}$$

$$|T_1^z(t) - T_2^z(t)| \leq H \qquad (T_1^x(t) - T_2^x(t))^2 + (T_1^y(t) - T_2^y(t))^2 \leq D^2$$

Example from Narkawicz, Muõz, Formal Verification of Conflict Detection Algorithms for Arbitrary Trajectories, 2012

Non-Linear Reasoning

Many applications



$$T_1^x(t) = 3.2484 + 270.7t + 433.12t^2 - 324.83999t^3$$

$$T_1^y(t) = 15.1592 + 108.28t + 121.2736t^2 - 649.67999t^3$$

$$T_1^z(t) = 38980.8 + 5414t - 21656t^2 + 32484t^3$$

$${}^2 2 + 3248.4t^3$$

$$36t^2 - 649.67999t^3$$

$$9816t^2 - 32484t^3$$

Run SMT solver

$$t \mapsto \frac{319}{16384} \approx 0.019470215$$

$D = 5$ $\quad H = 1000$ $\quad 0 \le t \le \frac{1}{20}$

$$|T_1^z(t) - T_2^z(t)| \le H \qquad (T_1^x(t) - T_2^x(t))^2 + (T_1^y(t) - T_2^y(t))^2 \le D^2$$

Example from Narkawicz, Muñoz, Formal Verification of Conflict Detection Algorithms for Arbitrary Trajectories, 2012

Non-Linear Reasoning

SMT techniques

Popular techniques in SMT (QF_NRA):

- Interval reasoning: RASAT
- Linear reasoning + model-based refinement: CVC4
- DPLL(T) + VTS: VERIT
- DPLL(T) + CAD: SMTRAT, VERIT
- MCSAT + CAD: Z3, YICES2

Non-Linear Reasoning

SMT techniques

Popular techniques in SMT (QF_NRA):

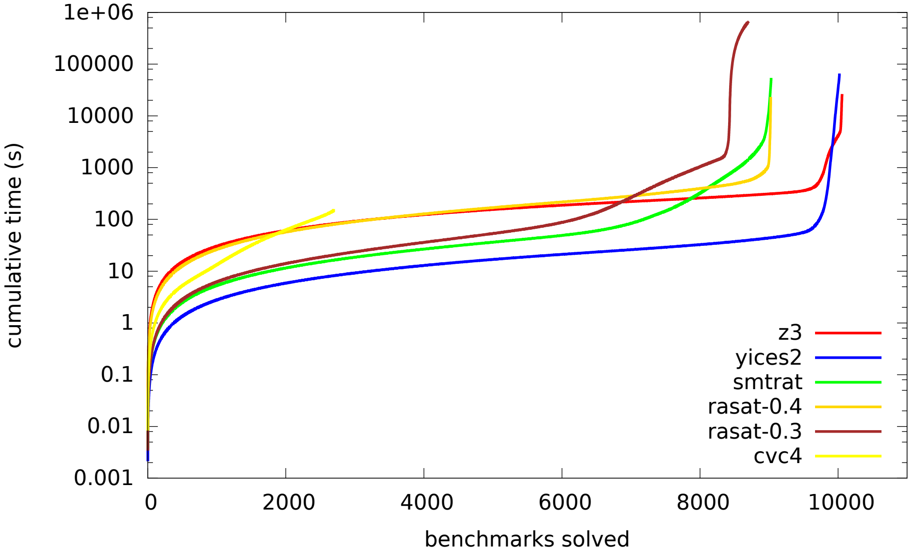
- Interval reasoning: RASAT
- Linear reasoning + model-based refinement: CVC4
- DPLL(T) + VTS: VERIT
- DPLL(T) + CAD: SMTRAT, VERIT
- MCSAT + CAD: Z3, YICES2

Cylindrical Algebraic Decomposition (CAD):

- complete method, currently state-of-the-art;
- requires advanced polynomial operations.

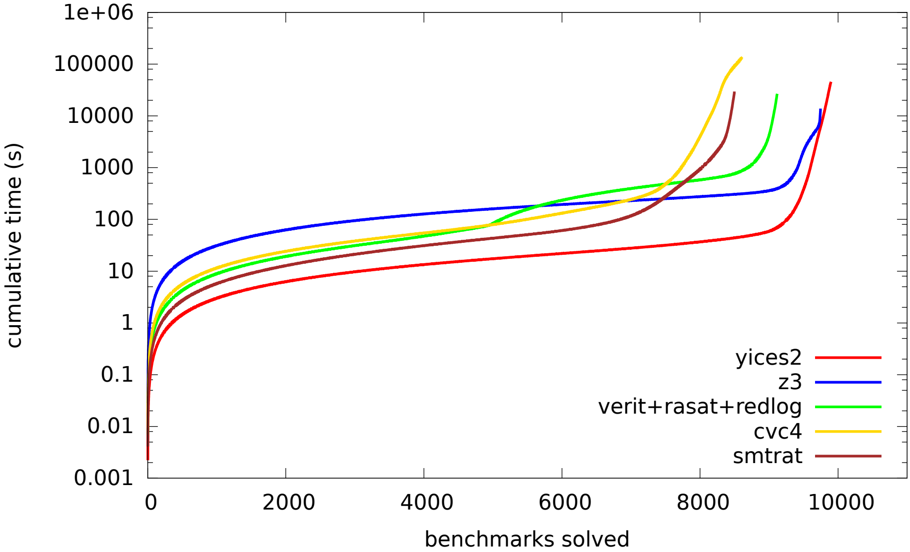
Non-Linear Reasoning

SMT solvers (2016)



Non-Linear Reasoning

SMT solvers (2017)



Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numbers.
4. Projection and symbolic explanations:
   - principal subresultant coefficients.

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system ☹

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanation
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system ☹
- Borrow and adapt code

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, . . .);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanation
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system ☹
- Borrow and adapt code ☹

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system ☹
- Borrow and adapt code ☹
- Implement yourself ☹

Non-Linear Reasoning

CAD-based reasoning

1. Representation of polynomials.
2. Basic operations:
   - variables, variable ordering;
   - arithmetic (addition, multiplication, ...);
   - GCD computation;
   - some factorization.
3. Solving and model representation:
   - Sturm sequences;
   - interval reasoning;
   - root isolation (multivariate);
   - resultants;
   - computation with algebraic numb
4. Projection and symbolic explanatior
   - principal subresultant coefficients.

How to get these?

- Use an existing library ☹
- Use a computer algebra system ☹
- Borrow and adapt code ☹
- Implement yourself ☹
- Use LibPoly ☺.

Outline

Introduction

LibPoly

- Working with Polynomials
- Constructing a Sign Table
- Cylindrical Algebraic Decomposition

Conclusion

LibPoly

- Open source: `https://github.com/SRI-CSL/libpoly`.
- Permissive License: LGLP
- Lightweight: Implemented in C, 15KLOC.
- Only depends on GMP.
- Basis for non-linear reasoning in YICES2.

Polynomial Basics

- Polynomials with coefficients over $\mathbb{Z}$.
- $\mathbb{Z}[x_1, \dots, x_n]$ are polynomials over variables $\vec{x} = \langle x_1, \dots, x_n \rangle$.
- For $f \in \mathbb{Z}[\vec{y}, x]$:

$$f(\vec{y}, x) = a_m \cdot x^{d_m} + a_{m-1} \cdot x^{d_{m-1}} + \cdots + a_1 \cdot x^{d_1} + a_0 \quad .$$

- $a_m \neq 0, a_i \in \mathbb{Z}[\vec{y}], d_m > \cdots > d_1 > 0$
- x is the top variable
- d_m is the degree of f
- a_m is the leading coefficient

Assignment and Evaluation

An assignment assigns variables to values

$$m = \{\, x \mapsto 1, y \mapsto 2, z \mapsto 3 \,\} \quad .$$

We can evaluate the sign of a polynomial $f \in \mathbb{Z}[x, y, z]$

$$\mathrm{sgn}(f, m) \in \{+1, 0, -1\} \quad .$$

Zeros of a Polynomial (Root Isolation)

Root Isolation

For $f \in \mathbb{Z}[\vec{y}, x]$ and an assignment $\vec{y} \mapsto \vec{v}$, find solutions to $f(\vec{v}, x) = 0$.

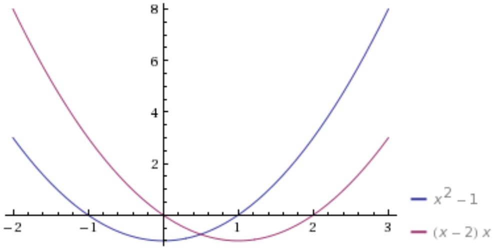
Example

$$m_1 = \{\} \qquad f_1(x) = x - 1$$

$$m_2 = \{x \mapsto 1\} \qquad f_2(x, y) = y^2 - 2x$$

$$m_3 = \{x \mapsto 1, y \mapsto \sqrt{2}\} \qquad f_3(x, y, z) = z^3 - y^2 - x$$

Sign Table



Example (Sign Table)

	$(-\infty, -1)$	$[-1]$	$(-1, 0)$	$[0]$	$(0, 1)$	$[1]$	$(1, 2)$	$[2]$	$(2, +\infty)$
$x^2 - 1$	+	0	-	-	-	0	+	+	+
$x(x - 2)$	+	+	+	0	-	-	-	0	+

Sign Table: What is it?

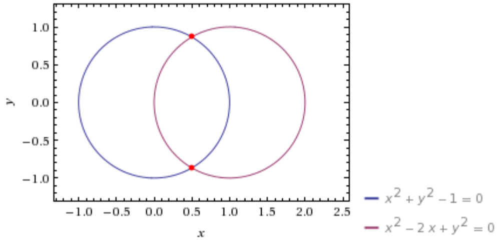
Example (Sign Table)

	$(-\infty, -1)$	$[-1]$	$(-1, 0)$	$[0]$	$(0, 1)$	$[1]$	$(1, 2)$	$[2]$	$(2, +\infty)$
$x^2 - 1$	+	0	-	-	-	0	+	+	+
$x(x-2)$	+	+	+	0	-	-	-	0	+

Sign Table

- Partition of $\mathbb{R}$ into intervals $I_1, \ldots I_n$.
- Picking an **arbitrary** sample value $v \in I_k$ is enough to evaluate signs.
- It completely characterizes the behavior of the polynomials.

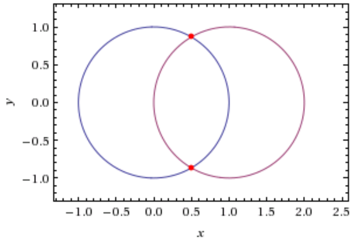
Can we do multivariate?



Example (Multivariate)

$$x^2 + y^2 - 1 \leq 0 \ , \qquad\qquad (x-1)^2 + y^2 - 1 \leq 0 \ .$$

Can we do multivariate?



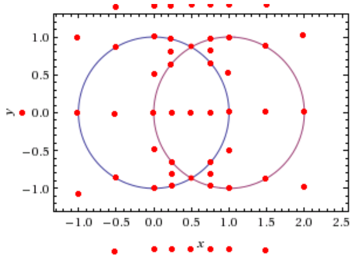
Recursive Sign Table

1. Pick order, say $x < y$.
2. P_x: polynomials in x.
3. P_y: polynomials in x, y.
4. Construct sign table T_x for P_x.
5. For each sample $v \in T_x$:
   - Construct sign table $T_{v,y}$ for P_y.

Example (Multivariate)

$$x^2 + y^2 - 1 \leq 0 \ , \qquad (x-1)^2 + y^2 - 1 \leq 0 \ .$$

Can we do multivariate?



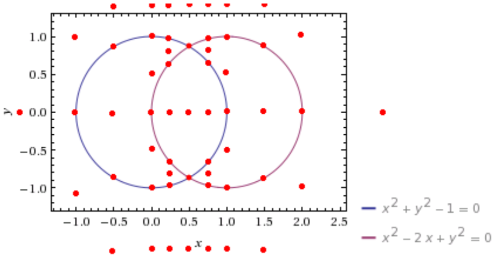
Recursive Sign Table

1. Pick order, say $x < y$.
2. P_x: polynomials in x.
3. P_y: polynomials in x, y.
4. Construct sign table T_x for P_x.
5. For each sample $v \in T_x$:
   - Construct sign table $T_{v,y}$ for P_y.

Example (Multivariate)

$$x^2 + y^2 - 1 \leq 0 \quad , \qquad\qquad (x-1)^2 + y^2 - 1 \leq 0 \quad .$$

Can we do multivariate?



Example (How to Get the Extra Polynomials?)

We added extra polynomials

$$x + 1 \ , \qquad x \ , \qquad 2x - 1 \ , \qquad x - 1 \ , \qquad x - 2 \ .$$

Can we find these polynomials automatically?

CAD Projection

Filling the blanks

Definition (Projection)

Given a set of polynomials $A = \{f_1, \ldots, f_m\} \subset \mathbb{Z}[\vec{y}, x]$, the x-projection of A is

$$P(A, x) = \bigcup_{f \in A} \mathbf{coeff}(f, x) \cup \bigcup_{\substack{f \in A \\ g \in \mathbf{R}^*(f,x)}} \mathbf{psc}(g, g'_x, x) \cup \bigcup_{\substack{i<j \\ g_i \in \mathbf{R}^*(f_i,x) \\ g_j \in \mathbf{R}^*(f_j,x)}} \mathbf{psc}(g_i, g_j, x) \quad .$$

CAD Projection

Filling the blanks

Definition (Projection)

Given a set of polynomials $A = \{f_1, \ldots, f_m\} \subset \mathbb{Z}[\vec{y}, x]$, the x-projection of A is

$$P(A, x) = \bigcup_{f \in A} \mathbf{coeff}(f, x) \cup \bigcup_{\substack{f \in A \\ g \in \mathbf{R}^*(f,x)}} \mathbf{psc}(g, g'_x, x) \cup \bigcup_{\substack{i<j \\ g_i \in \mathbf{R}^*(f_i,x) \\ g_j \in \mathbf{R}^*(f_j,x)}} \mathbf{psc}(g_i, g_j, x) \ .$$

coeff(f, x): Coefficients

Signs of coefficients invariant on $S \Rightarrow$ degrees of $f \in A$ invariant on S.

CAD Projection

Filling the blanks

Definition (Projection)

Given a set of polynomials $A = \{f_1, \ldots, f_m\} \subset \mathbb{Z}[\vec{y}, x]$, the x-projection of A is

$$P(A, x) = \bigcup_{f \in A} \textbf{coeff}(f, x) \cup \bigcup_{\substack{f \in A \\ g \in \mathbf{R^*}(f,x)}} \textbf{psc}(g, g'_x, x) \cup \bigcup_{\substack{i<j \\ g_i \in \mathbf{R^*}(f_i,x) \\ g_j \in \mathbf{R^*}(f_j,x)}} \textbf{psc}(g_i, g_j, x) \; .$$

$\mathsf{R}^*(f, x)$: Reductums include the "right degree" polynomials

$$f = \sum_{k=0}^{n} a_k x^k \;, \qquad \mathsf{R}(f, x) = \sum_{k=0}^{n-1} a_k x^k \;, \qquad \mathsf{R}^*(f, x) = \{f, \mathsf{R}(f), \mathsf{R}(\mathsf{R}(f)), \ldots\} \; .$$

CAD Projection

Filling the blanks

Definition (Projection)

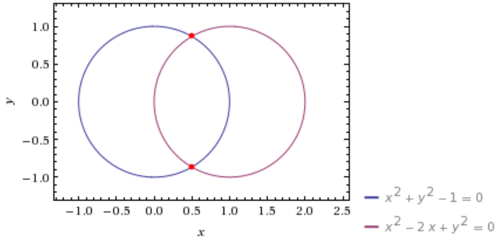
Given a set of polynomials $A = \{f_1, \ldots, f_m\} \subset \mathbb{Z}[\vec{y}, x]$, the x-projection of A is

$$P(A, x) = \bigcup_{f \in A} \textsf{coeff}(f, x) \cup \bigcup_{\substack{f \in A \\ g \in \textsf{R*}(f,x)}} \textsf{psc}(g, g'_x, x) \cup \bigcup_{\substack{i<j \\ g_i \in \textsf{R*}(f_i,x) \\ g_j \in \textsf{R*}(f_j,x)}} \textsf{psc}(g_i, g_j, x) \ .$$

Principal Subresultant Coefficients (PSC)

Signs of PSC invariant on S $\Rightarrow$ degree of $\gcd$ invariant on S.

CAD: Multivariate Sign Table

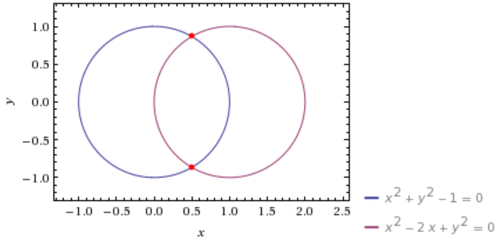


Projection: Extra Polynomials

Given a set of polynomials $A \subseteq \mathbb{Z}[x_1, \ldots, x_n]$:

- Project variable x_n.
- Project variable x_{n-1}.
- $\ldots$

CAD: Multivariate Sign Table

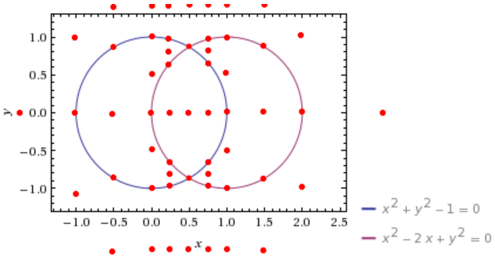


Lifting: Construct the Sign Table

Construct the table variable by variable:

- Isolate roots of x_1, pick a value in an interval.
- Isolate roots of x_2, pick a value in an interval.
- $\ldots$

CAD: Multivariate Sign Table

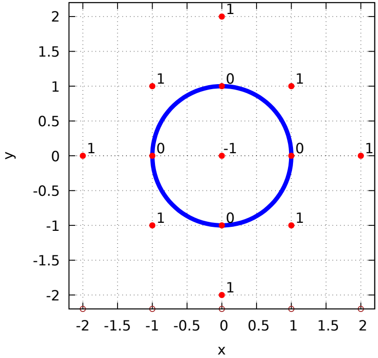


Lifting: Construct the Sign Table

Construct the table variable by variable:

- Isolate roots of x_1, pick a value in an interval.
- Isolate roots of x_2, pick a value in an interval.
- $\ldots$

CAD: Multivariate Sign Table



- Polynomial: $x^2 + y^2 - 1$.
- Projection: $x^2 - 1$.

Outline

Conclusion

A library for non-linear reasoning:

- Open source `https://github.com/SRI-CSL/libpoly`.
- Permissive License: LGLP
- Ubuntu and Brew packages incoming.
- Lightweight: Implemented in C, around 15KLOC.
- Only depends on GMP.
- Basis for non-linear reasoning in YICES2.
- Both for traditional CAD and MCSAT-style CAD.

Conclusion

A library for non-linear reasoning:

- Open source `https://github.com/SRI-CSL/libpoly`.
- Permissive License: LGLP
- Ubuntu and Brew packages incoming.
- Lightweight: Implemented in C, around 15KLOC.
- Only depends on GMP.
- Basis for non-linear reasoning in YICES2.
- Both for traditional CAD and MCSAT-style CAD.

Thank you!