fundamental ideas of Cylindrical Algebraic Decomposition

Gereon Kremer

fundamental solving approaches

let's review some basic ideas:

fundamental solving approaches

let's review some basic ideas:

- Boolean
  domain is finite, enumerate solutions
- bit-vectors, floating-point
  domain is finite, let the SAT solver figure it out
- uninterpreted functions
  congruence closure, number of arrangements is finite
- arrays
  reduce to uninterpreted functions on demand
- strings
  domain is finite but enumerable, use clever rewrites

fundamental solving approaches

let's review some basic ideas:

- Boolean
  domain is finite, enumerate solutions
- bit-vectors, floating-point
  domain is finite, let the SAT solver figure it out
- uninterpreted functions
  congruence closure, number of arrangements is finite
- arrays
  reduce to uninterpreted functions on demand
- strings
  domain is finite but enumerable, use clever rewrites
- integers
  assume bounds and reduce to bit-vectors, or rely on real solver

fundamental solving approaches

let's review some basic ideas:

- Boolean
  domain is finite, enumerate solutions
- bit-vectors, floating-point
  domain is finite, let the SAT solver figure it out
- uninterpreted functions
  congruence closure, number of arrangements is finite
- arrays
  reduce to uninterpreted functions on demand
- strings
  domain is finite but enumerable, use clever rewrites
- integers
  assume bounds and reduce to bit-vectors, or rely on real solver

what about real arithmetic?

solving for real arithmetic

fundamental problem: domain $\mathbb{R}$ is uncountably infinite

LRA stays in $\mathbb{Q}$ that is countable, but enumeration is not feasible

what is the CS way to deal with large search spaces?

solving for real arithmetic

fundamental problem: domain $\mathbb{R}$ is uncountably infinite

LRA stays in $\mathbb{Q}$ that is countable, but enumeration is not feasible

what is the CS way to deal with large search spaces?

abstraction!

solving for real arithmetic

fundamental problem: domain $\mathbb{R}$ is uncountably infinite
LRA stays in $\mathbb{Q}$ that is countable, but enumeration is not feasible

what is the CS way to deal with large search spaces?

abstraction!

general theme:

- look at the constraints, not at the solutions
- witness satisfiability in terms of the constraints
- the solution will show up as a by-product

Fourier-Motzkin

variable elimination for linear real arithmetic

$$\bigwedge_i a_i \leq x \wedge \bigwedge_j x \leq b_j \Rightarrow \bigwedge_{i,j} a_i \leq b_j$$

$$\begin{aligned}
& & 1 \leq x \wedge x \leq 7 - 2y \wedge x \leq 2y - 1 \\
\Rightarrow_x & & 1 \leq 7 - 2y \wedge 1 \leq 2y - 1 \\
\Rightarrow & & y \leq 3 \wedge 1 \leq y \\
\Rightarrow_y & & 1 \leq 3
\end{aligned}$$

construct model from the bottom, for example $y \mapsto 2, x \mapsto 2$

Fourier-Motzkin

variable elimination for linear real arithmetic

$$\bigwedge_i a_i \leq x \land \bigwedge_j x \leq b_j \Rightarrow \bigwedge_{i,j} a_i \leq b_j$$

$$\begin{aligned} & & 1 \leq x \land x \leq 7 - 2y \land x \leq 2y - 1 \\ \Rightarrow_x & & 1 \leq 7 - 2y \land 1 \leq 2y - 1 \\ \Rightarrow & & y \leq 3 \land 1 \leq y \\ \Rightarrow_y & & 1 \leq 3 \end{aligned}$$

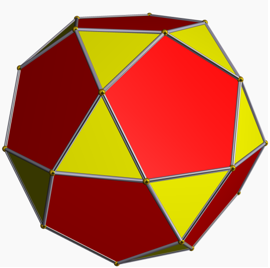
construct model from the bottom, for example $y \mapsto 2, x \mapsto 2$

- procedure only looks at constraints
- satisfiability is witnessed by $true$
- model construction is trivial

Simplex

optimization procedure for linear real arithmetic

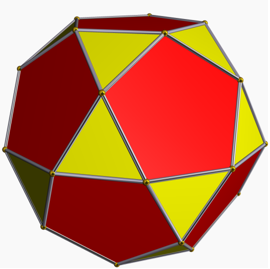
- linear constraint $=$ halfspace
- solution space is a polytope
  an intersection of halfspaces
- any corner of this polytope is a solution
- a corner is (uniquely) defined by the intersection of n halfspaces
- swap out one halfspace to reach neighbouring corner



Simplex

optimization procedure for linear real arithmetic

- linear constraint = halfspace
- solution space is a polytope
  an intersection of halfspaces
- any corner of this polytope is a solution
- a corner is (uniquely) defined by the intersection of n halfspaces
- swap out one halfspace to reach neighbouring corner

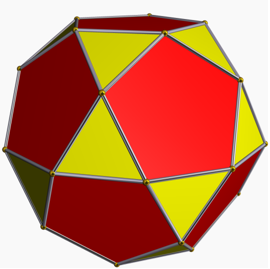


- only look at finite number of candidates (the corners)
- actual values are given by selection of constraints
- clever way to navigate these selections

Simplex

optimization procedure for linear real arithmetic

- linear constraint = halfspace
- solution space is a polytope
  an intersection of halfspaces
- any corner of this polytope is a solution
- a corner is (uniquely) defined by the intersection of n halfspaces
- swap out one halfspace to reach neighbouring corner



- only look at finite number of candidates (the corners)
- actual values are given by selection of constraints
- clever way to navigate these selections

for SMT: transform $\varphi \rightsquigarrow \varphi'$ such that $\overline{0} \models \varphi'$ and objective $o(\alpha) = 0$ ensures $\alpha \models \varphi$

virtual substitution

variable elimination by solution formulae

virtual substitution

variable elimination by solution formulae $x = -\frac{p}{2} \pm \sqrt{\frac{p}{2}^2 - q}$

virtual substitution

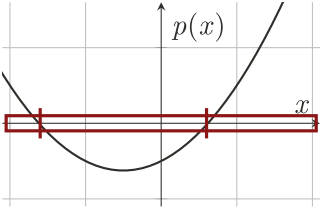
variable elimination by solution formulae $x = -\frac{p}{2} \pm \sqrt{\frac{p}{2}^2 - q}$

$$a \cdot x^2 + b \cdot x + c > 0 \qquad a, b, c \in \mathbb{Q}[\overline{y}]$$

virtual substitution

variable elimination by solution formulae $x = -\frac{p}{2} \pm \sqrt{\frac{p}{2}^2 - q}$

$$a \cdot x^2 + b \cdot x + c > 0 \qquad a, b, c \in \mathbb{Q}[\overline{y}]$$

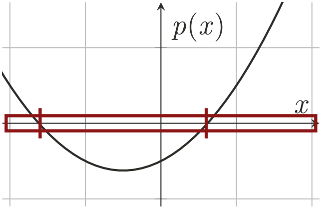


- α_i parametric roots for x
- abstract to $(\infty, \alpha_1), \alpha_1, (\alpha_1, \alpha_2), \ldots$
- multiple constraints: α_k are parametric
- core idea: symbolic $-\infty$ and $\alpha_k + \varepsilon$
- one test candidate per interval: $-\infty, \alpha_1, \alpha_1 + \varepsilon, \ldots$

virtual substitution

variable elimination by solution formulae $x = -\frac{p}{2} \pm \sqrt{\frac{p}{2}^2 - q}$

$$a \cdot x^2 + b \cdot x + c > 0 \qquad a, b, c \in \mathbb{Q}[\overline{y}]$$



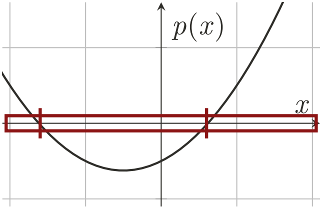
- α_i parametric roots for x
- abstract to $(\infty, \alpha_1), \alpha_1, (\alpha_1, \alpha_2), \dots$
- multiple constraints: α_k are parametric
- core idea: symbolic $-\infty$ and $\alpha_k + \varepsilon$
- one test candidate per interval: $-\infty, \alpha_1, \alpha_1 + \varepsilon, \dots$

- abstract from real intervals to representatives
- representatives are symbolic (in remaining variables)
- clever way to substitute symbolic values

virtual substitution

variable elimination by solution formulae $\qquad x = -\frac{p}{2} \pm \sqrt{\frac{p}{2}^2 - q}$

$$a \cdot x^2 + b \cdot x + c > 0 \qquad a, b, c \in \mathbb{Q}[\overline{y}]$$



- α_i parametric roots for x
- abstract to $(\infty, \alpha_1), \alpha_1, (\alpha_1, \alpha_2), \ldots$
- multiple constraints: α_k are parametric
- core idea: symbolic $-\infty$ and $\alpha_k + \varepsilon$
- one test candidate per interval: $-\infty, \alpha_1, \alpha_1 + \varepsilon, \ldots$

- abstract from real intervals to representatives
- representatives are symbolic (in remaining variables)
- clever way to substitute symbolic values

what about existence of solution formulae?

towards a general procedure

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$

towards a general procedure

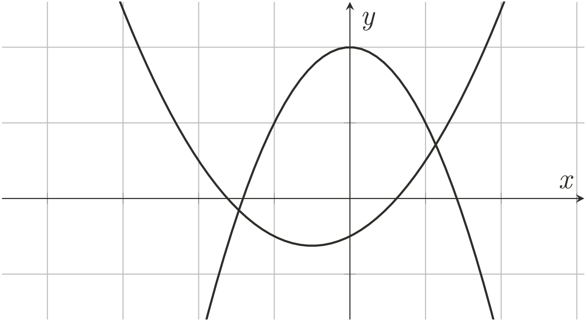
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$

$$\varphi := y > \frac{x^2 + x - 1}{2} \land y < 2 - x^2$$

towards a general procedure

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$

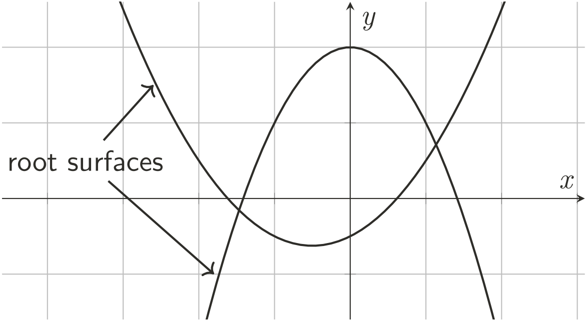
$$\varphi := y > \frac{x^2 + x - 1}{2} \wedge y < 2 - x^2$$



towards a general procedure

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$

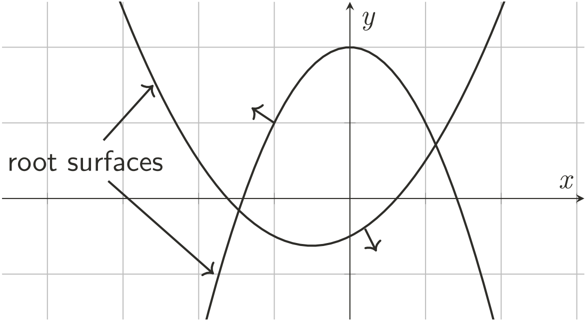
$$\varphi := y > \frac{x^2 + x - 1}{2} \land y < 2 - x^2$$



towards a general procedure

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$

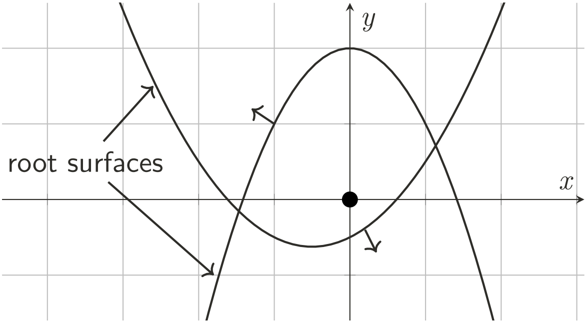
$$\varphi := y > \frac{x^2 + x - 1}{2} \wedge y < 2 - x^2$$



towards a general procedure

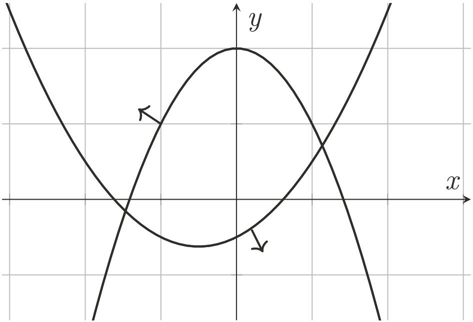
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$

$$\varphi := y > \frac{x^2 + x - 1}{2} \land y < 2 - x^2$$



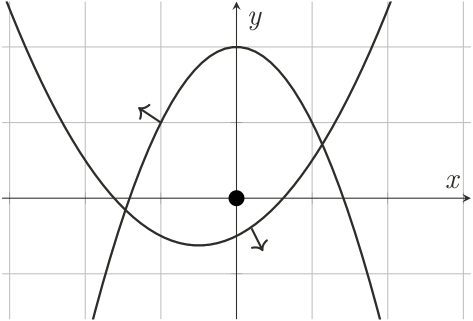
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



abstraction by sign-invariance

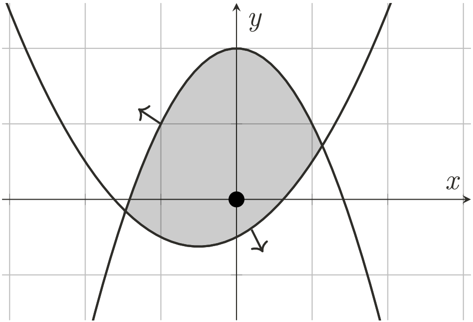
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- $(0,0) \models \varphi$

abstraction by sign-invariance

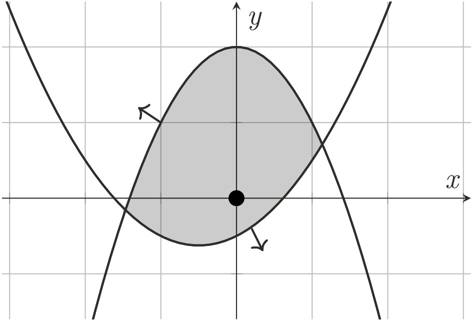
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- $(0, 0) \models \varphi$
- region around $(0, 0) \models \varphi$

abstraction by sign-invariance

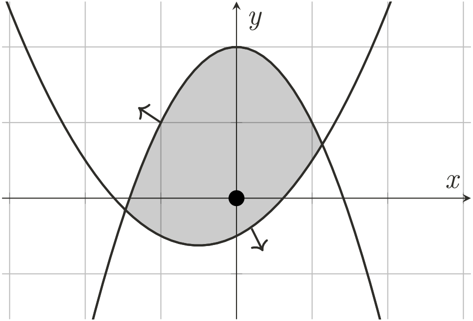
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



- $(0,0) \vDash \varphi$
- region around $(0,0) \vDash \varphi$, bounded by $roots(polys(\varphi))$

abstraction by sign-invariance

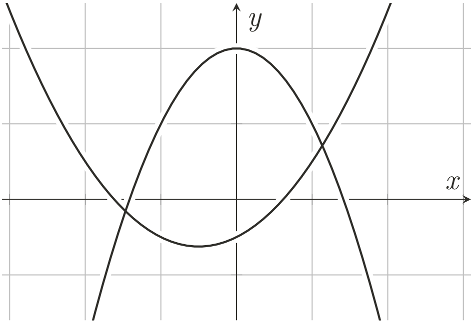
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



- $(0,0) \models \varphi$
- region around $(0,0) \models \varphi$, bounded by $roots(polys(\varphi))$
- sign-invariance $\Rightarrow$ truth-invariance

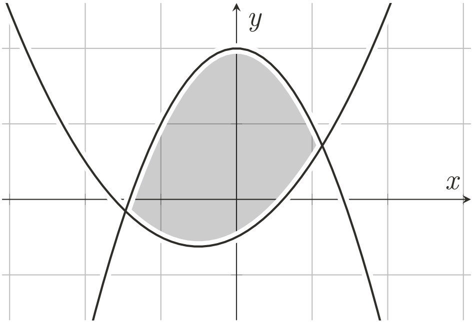
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



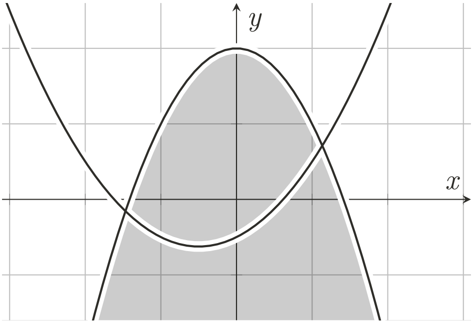
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



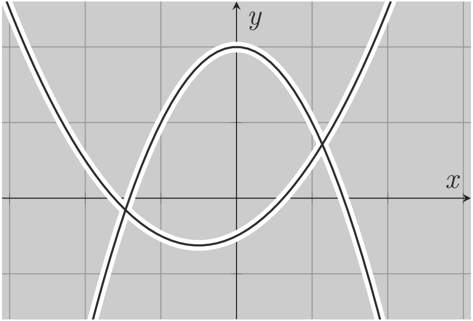
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



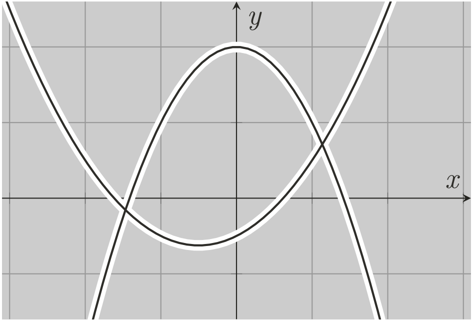
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



abstraction by sign-invariance

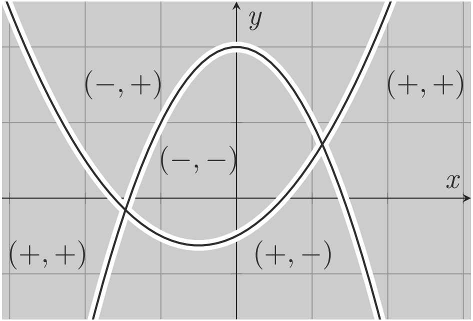
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- regions correspond to sign combinations

abstraction by sign-invariance

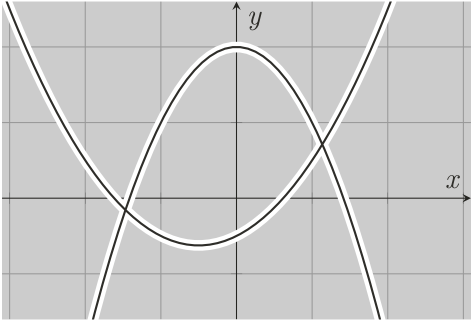
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- regions correspond to sign combinations

abstraction by sign-invariance

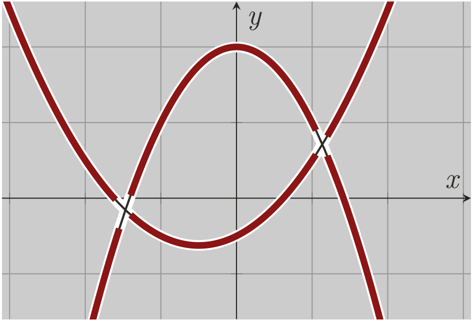
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- regions correspond to sign combinations

abstraction by sign-invariance

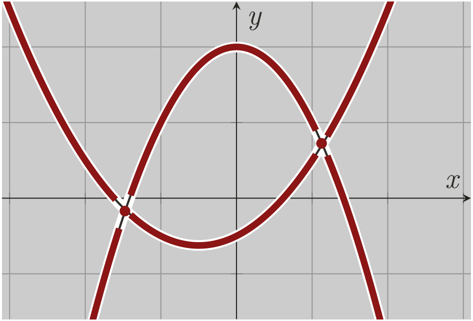
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



- regions correspond to sign combinations

abstraction by sign-invariance

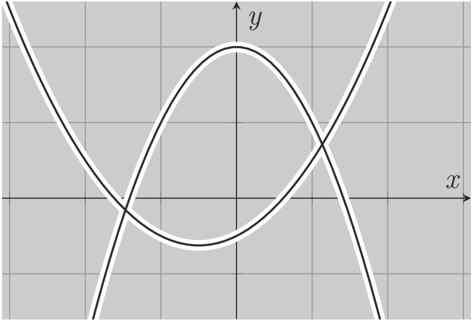
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- regions correspond to sign combinations

abstraction by sign-invariance

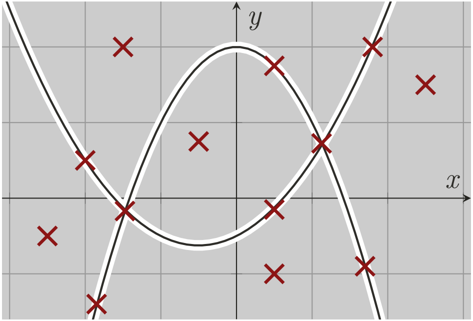
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \land x^2 + y - 2 < 0$$



- regions correspond to sign combinations
- abstract from region to sample points

abstraction by sign-invariance

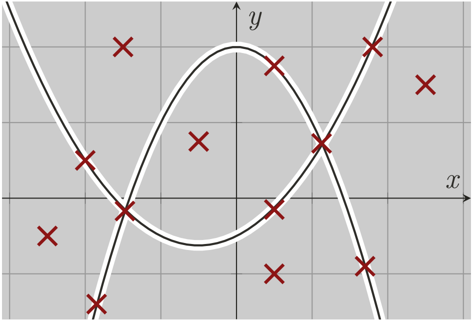
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



- regions correspond to sign combinations
- abstract from region to sample points

abstraction by sign-invariance

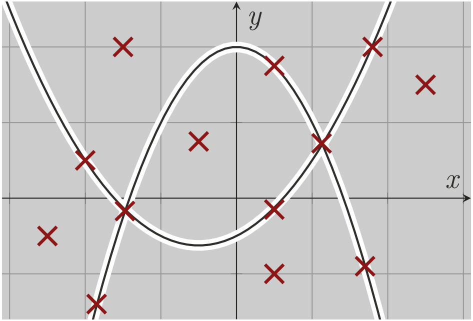
$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$



- regions correspond to sign combinations
- abstract from region to sample points
- abstract from $\mathbb{R}^n$ to finite set of sample points

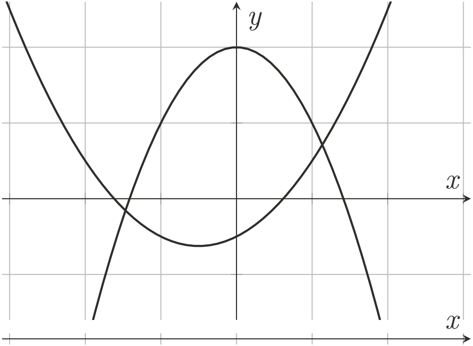
abstraction by sign-invariance

$$\varphi := x^2 + x - 1 - 2 \cdot y < 0 \wedge x^2 + y - 2 < 0$$

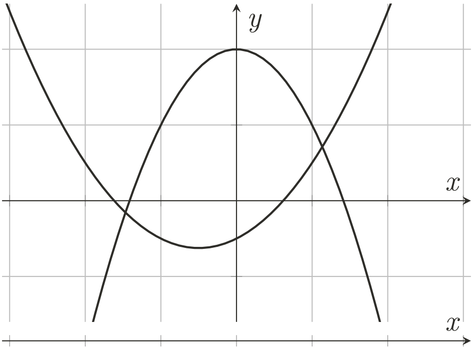


- regions correspond to sign combinations
- abstract from region to sample points
- abstract from $\mathbb{R}^n$ to finite set of sample points
- φ satisfiable $\Leftrightarrow$ there is a satisfying sample point

finding sample points

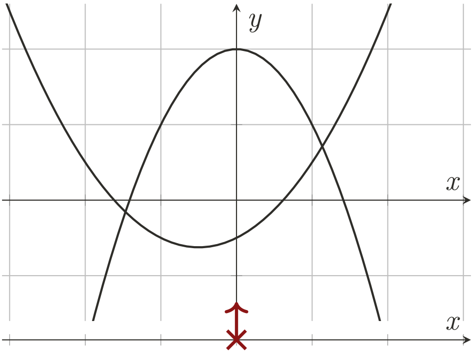


finding sample points



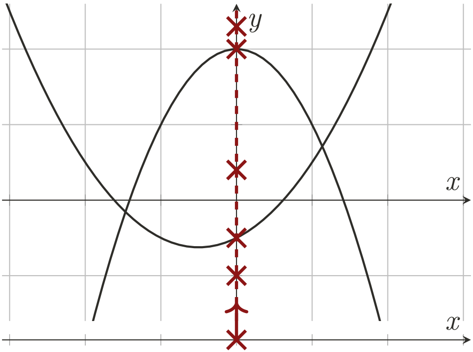
- one dimension at a time

finding sample points



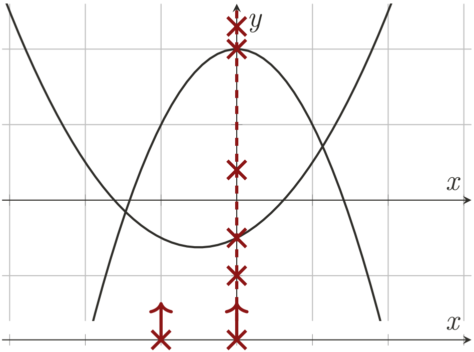
- one dimension at a time

finding sample points



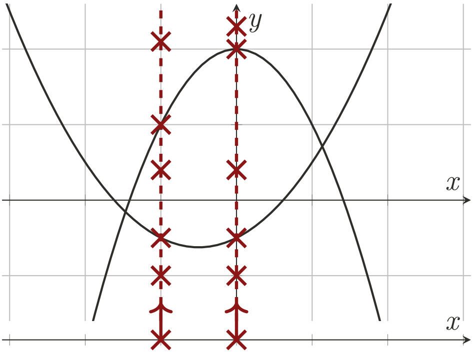
- one dimension at a time

finding sample points



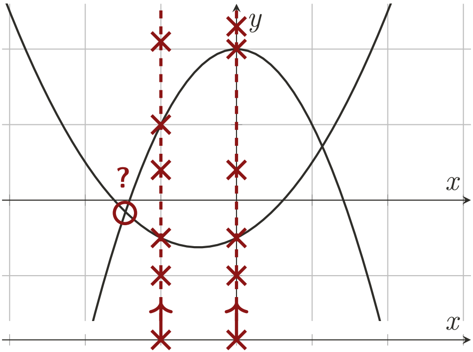
- one dimension at a time

finding sample points



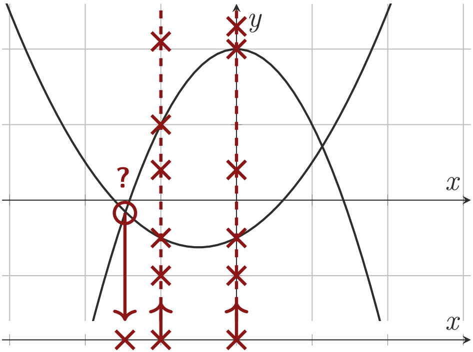
- one dimension at a time

finding sample points



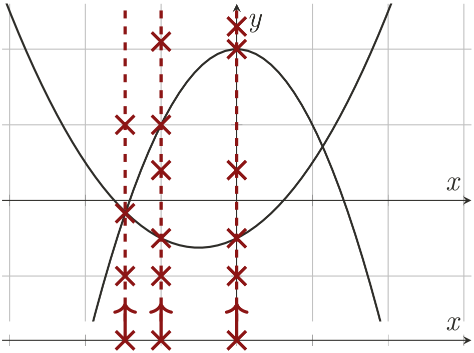
- one dimension at a time
- what about special points?

finding sample points



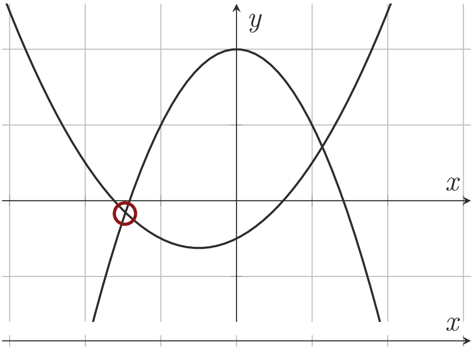
- one dimension at a time
- what about special points?
- project to lower dimension

finding sample points

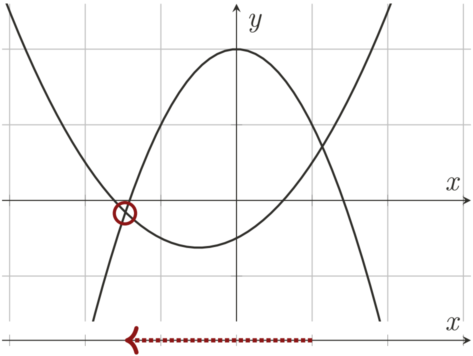


- one dimension at a time
- what about special points?
- project to lower dimension
- construct samples from these projections

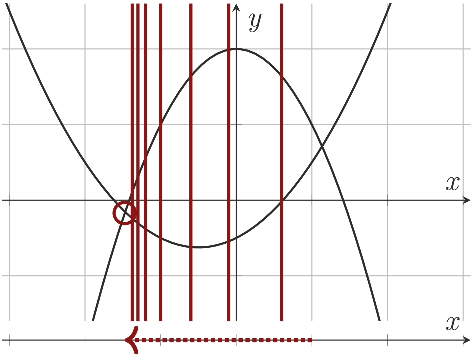
cylinders



cylinders

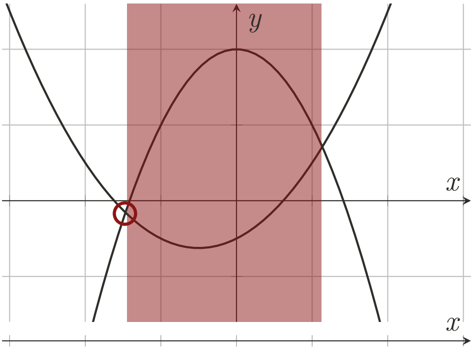


cylinders



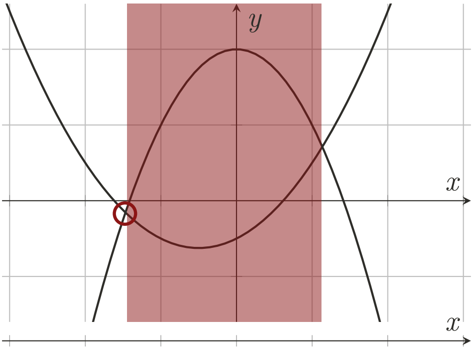
- arrangement of roots changes

cylinders



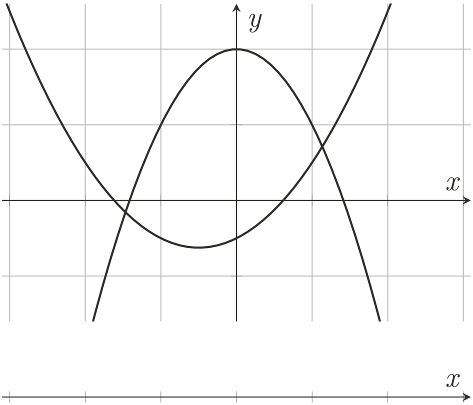
- arrangement of roots changes
- within a "cylinder": cylindrical arrangement of cells

cylinders

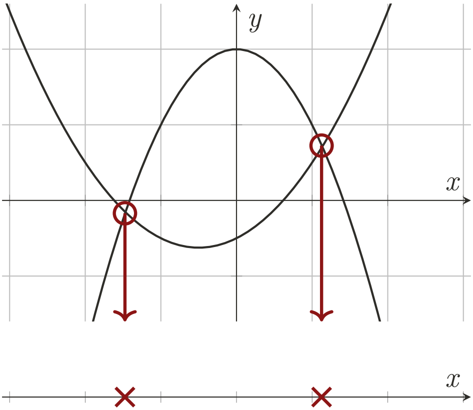


- arrangement of roots changes
- within a "cylinder": cylindrical arrangement of cells
- roots are delineable within a cylinder
- need to identify cylinder boundaries

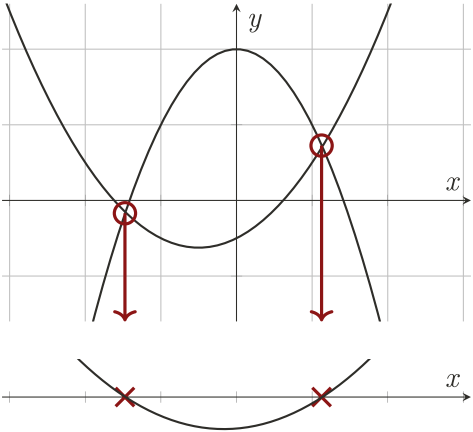
projecting cylinder boundaries – part 1



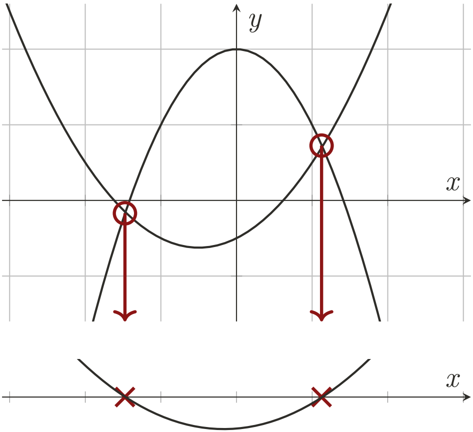
projecting cylinder boundaries – part 1



projecting cylinder boundaries – part 1

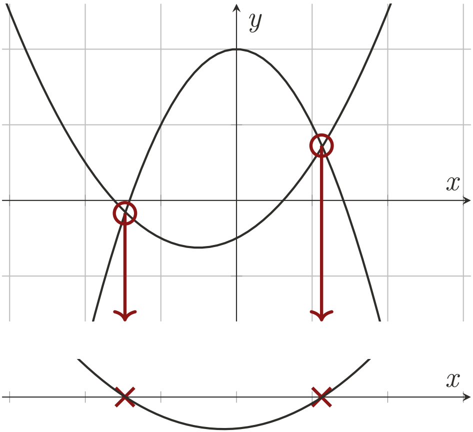


projecting cylinder boundaries – part 1



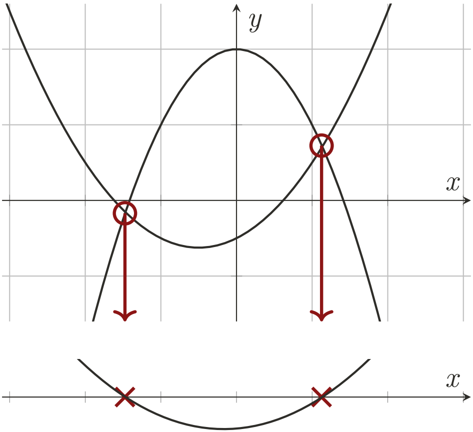
- $res_y(x^2 + x - 1 - 2 \cdot y, x^2 + y - 2) = 3 \cdot x^2 + x - 5$

projecting cylinder boundaries – part 1



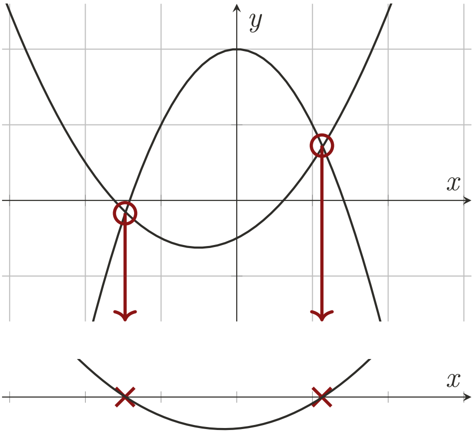
- $res_y(x^2 + x - 1 - 2 \cdot y, x^2 + y - 2) = 3 \cdot x^2 + x - 5$
- $\forall x, y. p(x, y) = q(x, y) = 0 \Rightarrow res_y(p, q)(x) = 0$

projecting cylinder boundaries – part 1



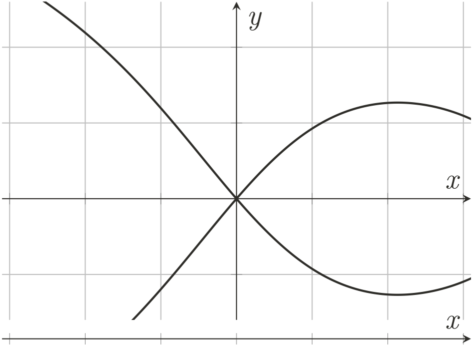
- $res_y(x^2 + x - 1 - 2 \cdot y, x^2 + y - 2) = 3 \cdot x^2 + x - 5$
- $\forall \overline{x}, y. p(\overline{x}, y) = q(\overline{x}, y) = 0 \Rightarrow res_y(p, q)(\overline{x}) = 0$

projecting cylinder boundaries – part 1

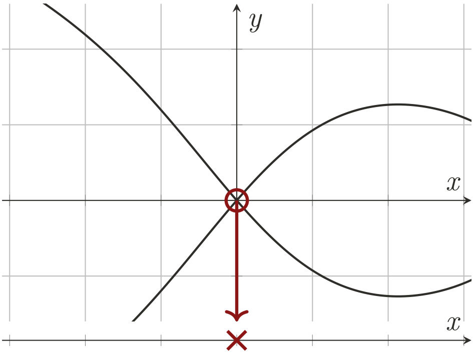


- $res_y(x^2 + x - 1 - 2 \cdot y, x^2 + y - 2) = 3 \cdot x^2 + x - 5$
- resultants indicate common roots of two polynomials

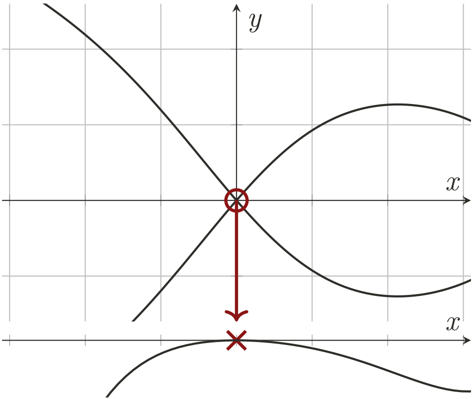
projecting cylinder boundaries – part 2



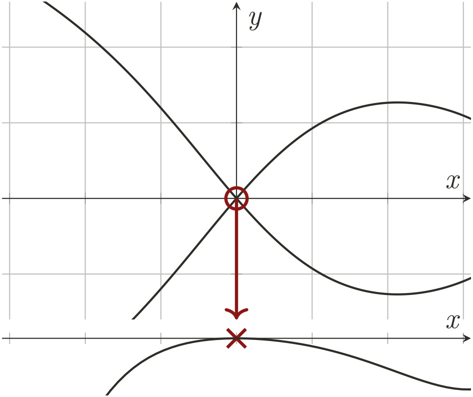
projecting cylinder boundaries – part 2



projecting cylinder boundaries – part 2

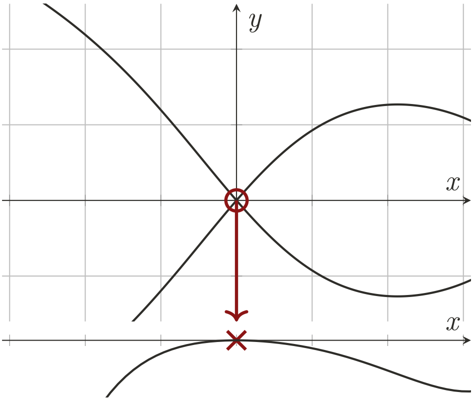


projecting cylinder boundaries – part 2



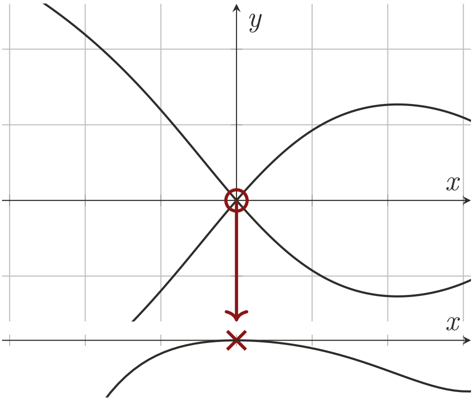
- $disc_y(x^3 + 0.5x^2y^2 - 4x^2 + 3y^3) = x^5 - 4x^4 + 6x^3 - 24x^2$

projecting cylinder boundaries – part 2



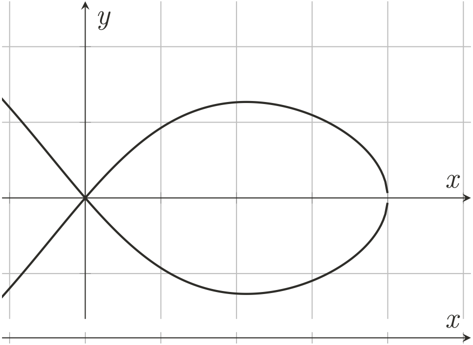
- $disc_y(x^3 + 0.5x^2y^2 - 4x^2 + 3y^3) = x^5 - 4x^4 + 6x^3 - 24x^2$
- $\forall \overline{x}, y. p(\overline{x}, y) = p'(\overline{x}, y) = 0 \Rightarrow disc_y(p, q)(\overline{x}) = 0$

projecting cylinder boundaries – part 2

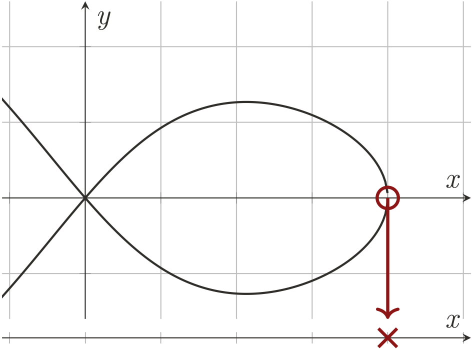


- $disc_y(x^3 + 0.5x^2y^2 - 4x^2 + 3y^3) = x^5 - 4x^4 + 6x^3 - 24x^2$
- $\forall \overline{x}, y. p(\overline{x}, y) = p'(\overline{x}, y) = 0 \Rightarrow disc_y(p, q)(\overline{x}) = 0$
- $disc_y(p) := res_y(p, p')$

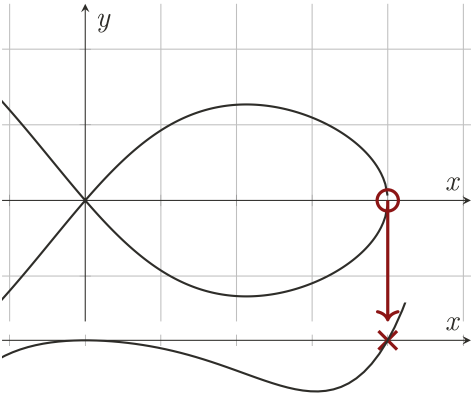
projecting cylinder boundaries – part 2



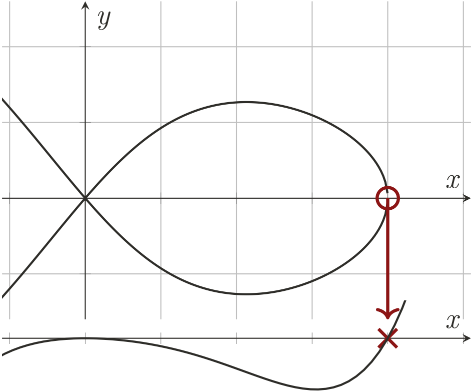
projecting cylinder boundaries – part 2



projecting cylinder boundaries – part 2

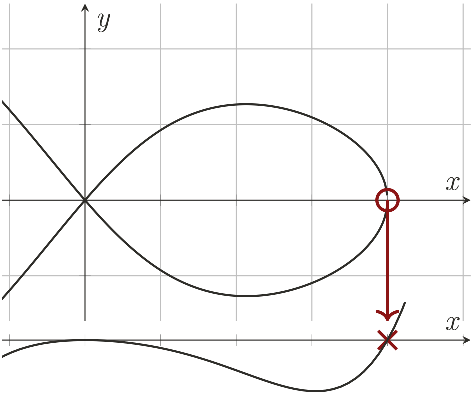


projecting cylinder boundaries – part 2



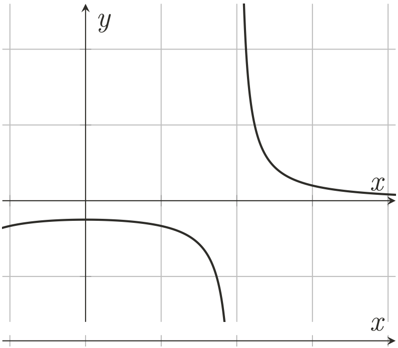
- $disc_y(x^3 + 0.5x^2y^2 - 4x^2 + 3y^3) = x^5 - 4x^4 + 6x^3 - 24x^2$
- $disc_y(p) \coloneqq res_y(p, p')$

projecting cylinder boundaries – part 2

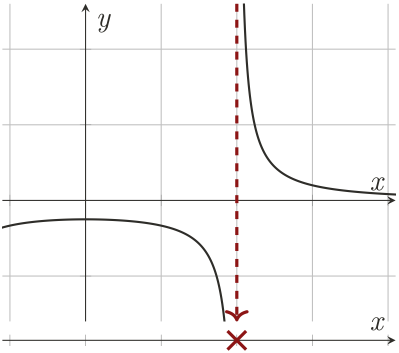


- $disc_y(x^3 + 0.5x^2y^2 - 4x^2 + 3y^3) = x^5 - 4x^4 + 6x^3 - 24x^2$
- $disc_y(p) := res_y(p, p')$
- discriminants indicate multiple roots of a single polynomial

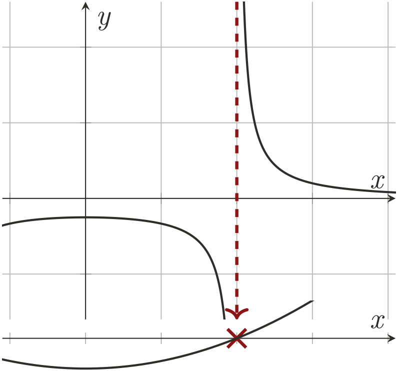
projecting cylinder boundaries – part 3



projecting cylinder boundaries – part 3

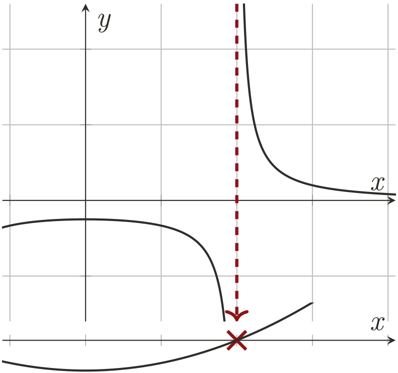


projecting cylinder boundaries – part 3



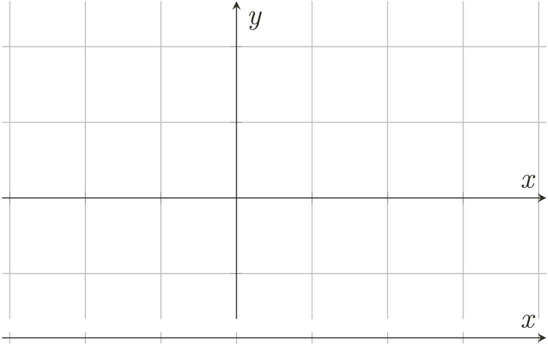
- $coeffs(x^2 y - 4y - 1) = \{x^2 - 4\}$

projecting cylinder boundaries – part 3



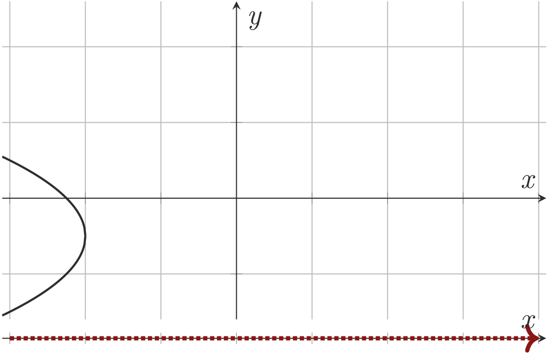
- $\mathit{coeffs}(x^2y - 4y - 1) = \{x^2 - 4\}$
- coefficients indicate singularities of a polynomial

projecting cylinder boundaries



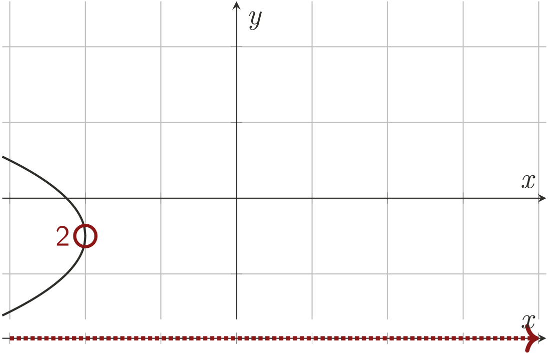
- (1) resultants (2) discriminants (3) coefficients

projecting cylinder boundaries



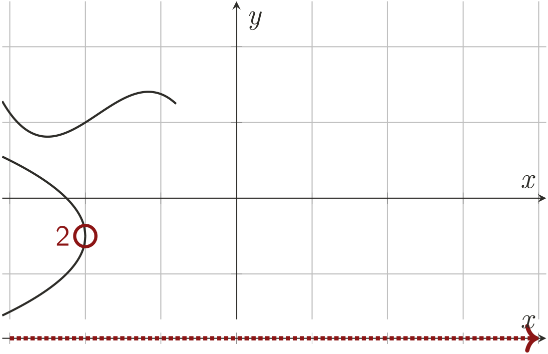
- (1) resultants (2) discriminants (3) coefficients

projecting cylinder boundaries



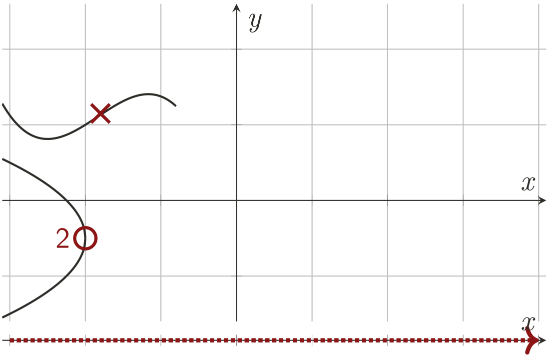
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse

projecting cylinder boundaries



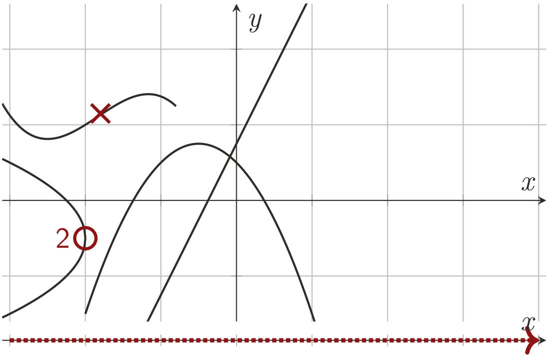
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse

projecting cylinder boundaries



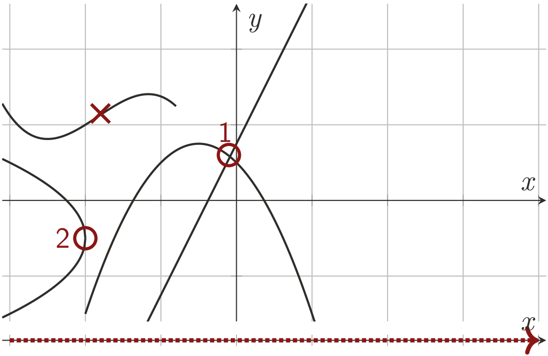
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse

projecting cylinder boundaries



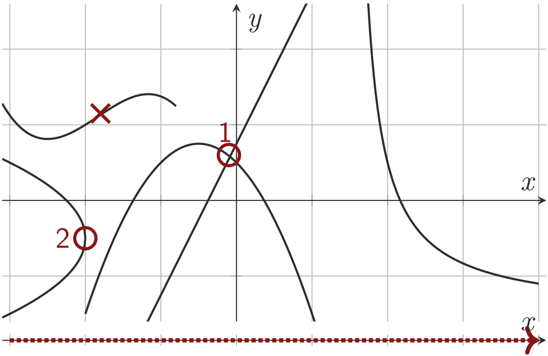
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse

projecting cylinder boundaries



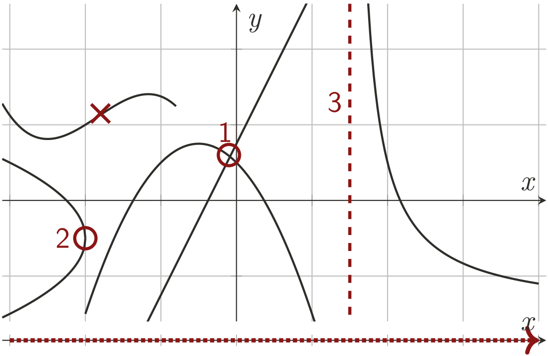
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse, change order

projecting cylinder boundaries



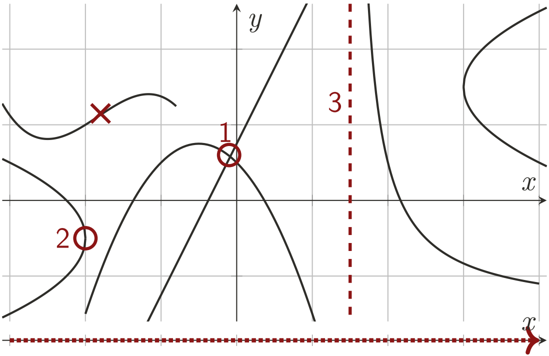
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse, change order

projecting cylinder boundaries



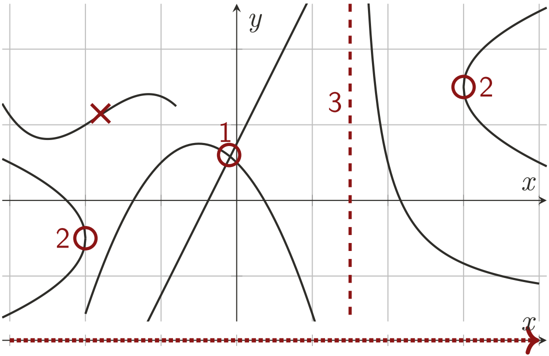
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse, change order, go to $\pm\infty$

projecting cylinder boundaries



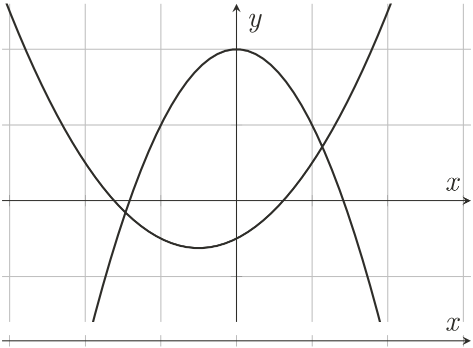
- (1) resultants (2) discriminants (3) coefficients
- roots can collapse, change order, go to $\pm\infty$

projecting cylinder boundaries

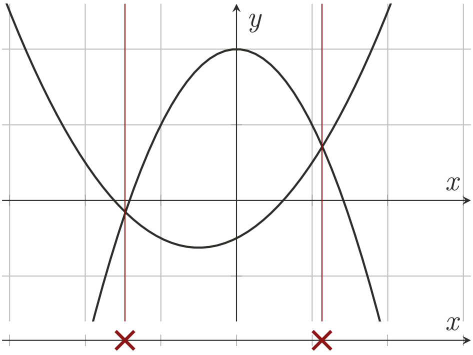


- (1) resultants (2) discriminants (3) coefficients
- roots can collapse, change order, go to $\pm\infty$, emerge

algorithmic idea

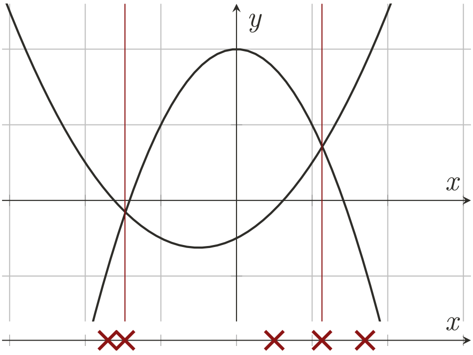


algorithmic idea



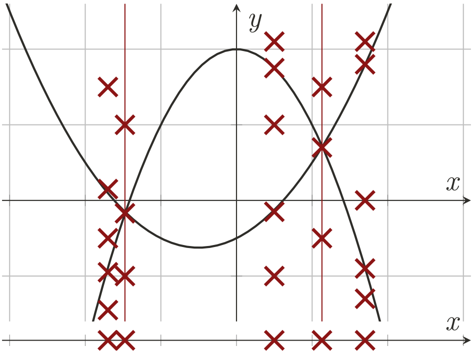
- project all cylinder boundaries

algorithmic idea



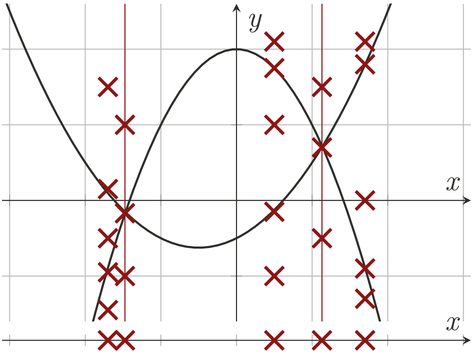
- project all cylinder boundaries
- construct one-dimensional samples

algorithmic idea



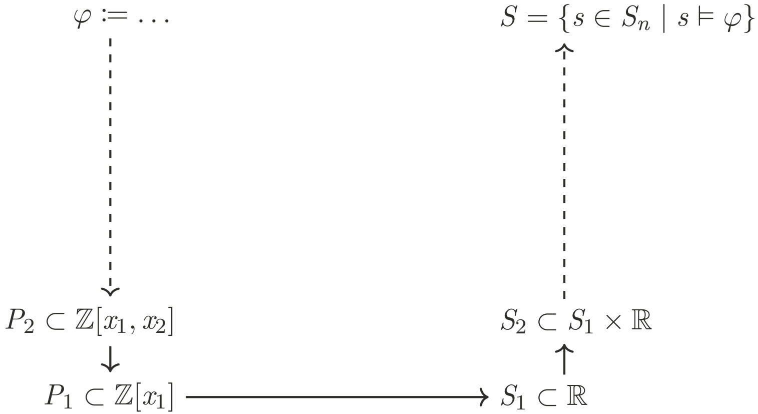
- project all cylinder boundaries
- construct one-dimensional samples
- lift to two-dimensional samples

algorithmic idea

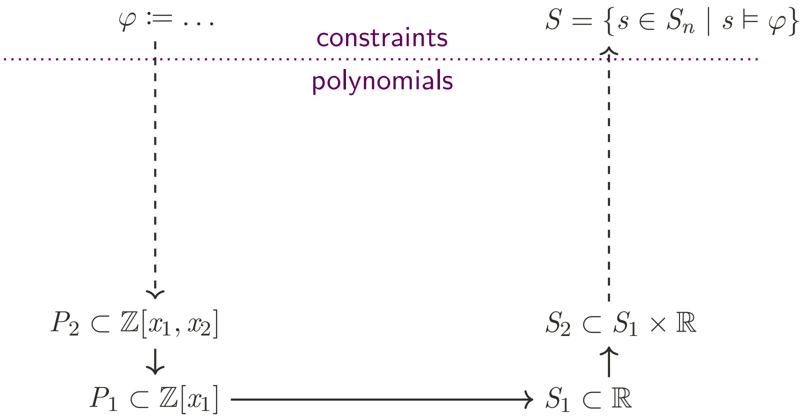


- project all cylinder boundaries — resultants, discriminants, coefficients
- construct one-dimensional samples — real root isolation
- lift to two-dimensional samples — real root isolation with partial model

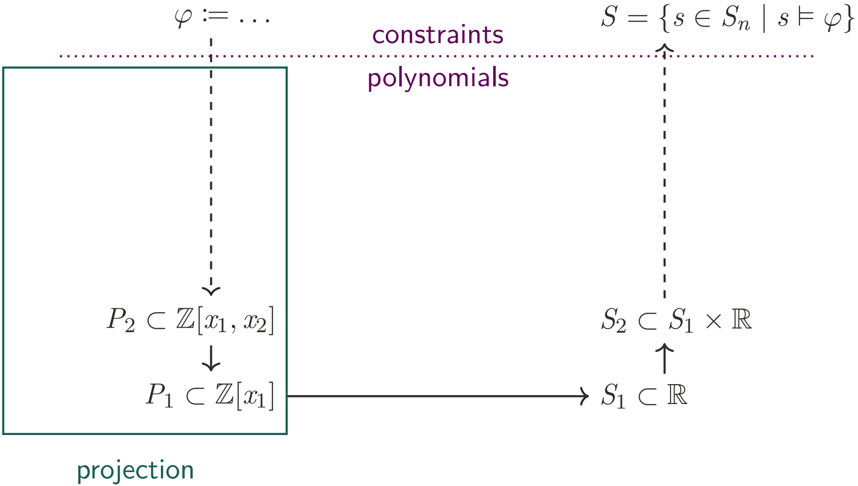
higher dimensions



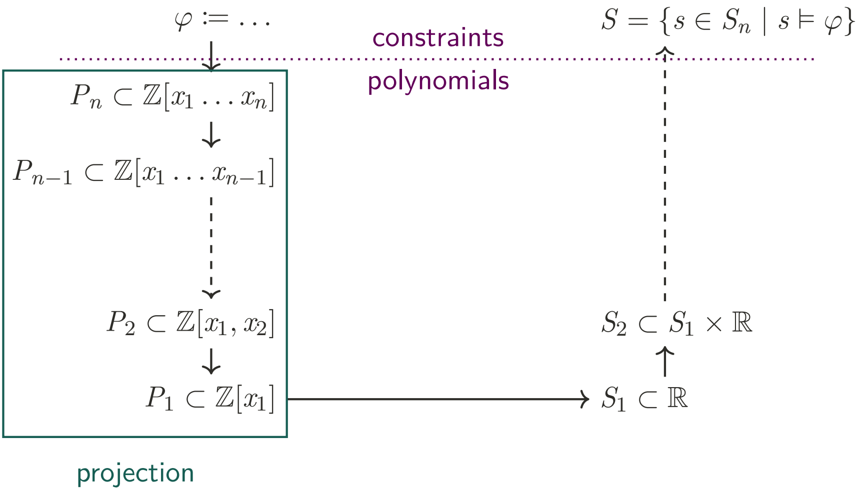
higher dimensions



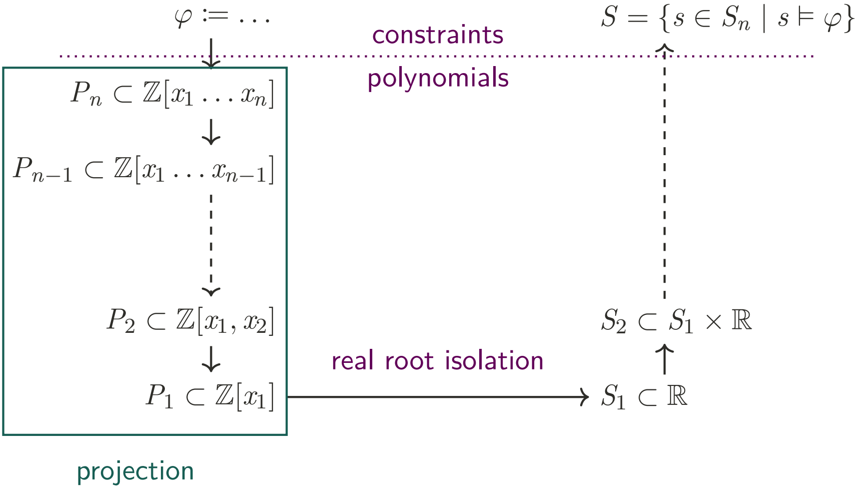
higher dimensions



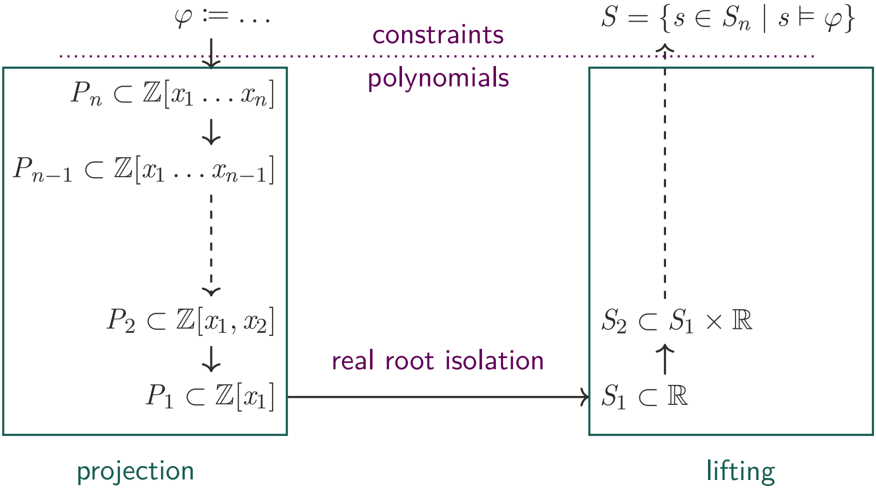
higher dimensions



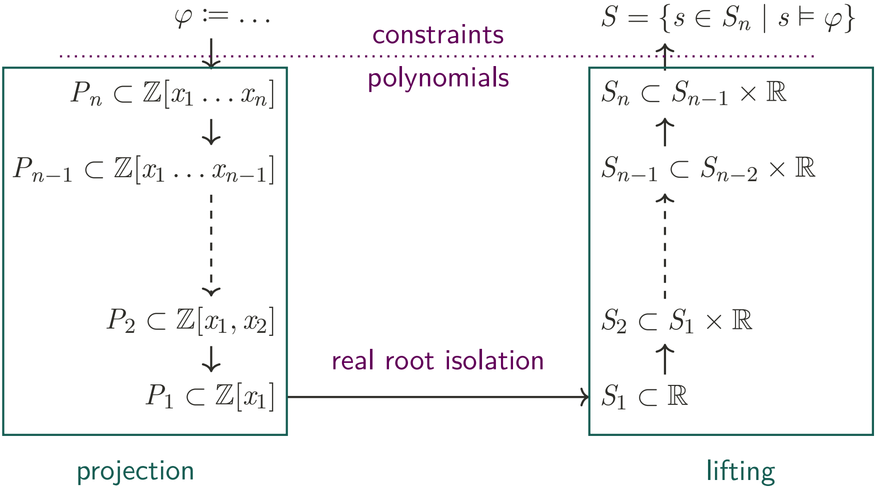
higher dimensions



higher dimensions



higher dimensions



bits and pieces

- polynomials may not have all variables $p \in \mathbb{Z}[x_1, x_n]$
- polynomials may nullify $p \in \mathbb{Z}[x, y],\ p(\alpha_x) = 0$

  $\rightarrow$ different projection operators, Lazard's lifting schema
- resultants may nullify $res(p \cdot q, q \cdot r) = 0$

  $\rightarrow$ factorize polynomials
- underlying machinery

  polynomials, real algebraic numbers, resultants, factorization, ...

  $\rightarrow$ `libpoly`, `CArL`, `CoCoALib`, CAS
- SMT compliancy (incrementality, backtracking, unsat cores)

bits and pieces

- polynomials may not have all variables $p \in \mathbb{Z}[x_1, x_n]$
- polynomials may nullify $p \in \mathbb{Z}[x, y],\ p(\alpha_x) = 0$
  $\rightarrow$ different projection operators, Lazard's lifting schema
- resultants may nullify $res(p \cdot q, q \cdot r) = 0$
  $\rightarrow$ factorize polynomials
- underlying machinery
  polynomials, real algebraic numbers, resultants, factorization, ...
  $\rightarrow$ `libpoly`, `CArL`, `CoCoALib`, CAS
- SMT compliancy (incrementality, backtracking, unsat cores)

thanks for your time!

literature

- other techniques [Fourier 1825] [Fourier 1826] [Dines 1919] [Motzkin 1936]; [Wolfman et al. 1999] [Dutertre et al. 2006] [Moura et al. 2008]; [Weispfenning 1997] [Kota et al. 2015]
- CAD [Collins 1974] [Arnon et al. 1984] [Davenport et al. 1988] [Caviness et al. 1998] [Collins 1998] [Bradford et al. 2016]
- projection [Collins 1974] [Hong 1990] [McCallum 1984] [Lazard 1994] [McCallum 1999] [Brown 2001] [England et al. 2015] [McCallum et al. 2016] [Haehn 2018] [McCallum et al. 2019]
- lifting [Collins et al. 1976] [Lazard 1994] [Kremer et al. 2021]
- adaptions / extensions [Jovanovi et al. 2012] [Moura et al. 2013] [Loup et al. 2013] [Brown et al. 2015] [Brown 2015] [Jaroschek et al. 2015] [Nalbach et al. 2019] [Kremer et al. 2020] [Ábrahám et al. 2021]
- implementations / tools [Brown 2003] [Chen et al. 2009] [Corzilius et al. 2015] [Jovanovic et al. 2017] [Abbott et al. 2018]
- heuristics [England et al. 2014] [Huang et al. 2014] [Kremer 2020]

References I

- John Abbott, Anna M. Bigatti, and Elisa Palezzato. "New in CoCoA-5.2.4 and CoCoALib-0.99600 for SC-Square". In: SC^2. FLoC. Vol. 2189. July 2018, pp. 88–94. URL: http://ceur-ws.org/Vol-2189/paper4.pdf.
- Erika Ábrahám, James H. Davenport, Matthew England, and Gereon Kremer. "Deciding the Consistency of Non-Linear Real Arithmetic Constraints with a Conflict Driven Search Using Cylindrical Algebraic Coverings". In: Journal of Logical and Algebraic Methods in Programming 119 (2021), p. 100633. DOI: 10.1016/j.jlamp.2020.100633.
- Dennis S. Arnon, George E. Collins, and Scott McCallum. "Cylindrical Algebraic Decomposition I: The Basic Algorithm". In: SIAM Journal on Computing 13 (4 1984), pp. 865–877. DOI: 10.1137/0213054.
- Russell Bradford, James H. Davenport, Matthew England, Scott McCallum, and David Wilson. "Truth table invariant cylindrical algebraic decomposition". In: Journal of Symbolic Computation 76 (2016), pp. 1–35. DOI: 10.1016/j.jsc.2015.11.002.
- Christopher W. Brown. "Improved Projection for Cylindrical Algebraic Decomposition". In: Journal of Symbolic Computation 32 (5 2001), pp. 447–465. DOI: 10.1006/jsco.2001.0463.
- Christopher W. Brown. "QEPCAD B: A program for computing with semi-algebraic sets using CADs". In: ACM SIGSAM Bulletin 37 (4 2003), pp. 97–108. DOI: 10.1145/968708.968710.

References II

- Christopher W. Brown. "Open Non-uniform Cylindrical Algebraic Decompositions". In: ISSAC. 2015, pp. 85–92. DOI: 10.1145/2755996.2756654.
- Christopher W. Brown and Marek Kota. "Constructing a single cell in cylindrical algebraic decomposition". In: Journal of Symbolic Computation 70 (2015), pp. 14–48. DOI: 10.1016/j.jsc.2014.09.024.
- Bob Forrester Caviness and Jeremy R. Johnson. Quantifier Elimination and Cylindrical Algebraic Decomposition. 1998. DOI: 10.1007/978-3-7091-9459-1.
- Changbo Chen, Marc Moreno Maza, Bican Xia, and Lu Yang. "Computing Cylindrical Algebraic Decomposition via Triangular Decomposition". In: ISSAC. 2009, pp. 95–102. DOI: 10.1145/1576702.1576718.
- George E. Collins. "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition–Preliminary Report". In: ACM SIGSAM Bulletin 8 (3 1974), pp. 80–90. DOI: 10.1145/1086837.1086852.
- George E. Collins. "Quantifier Elimination by Cylindrical Algebraic Decomposition — Twenty Years of Progress". In: Quantifier Elimination and Cylindrical Algebraic Decomposition. 1998, pp. 8–23. DOI: 10.1007/978-3-7091-9459-1_2.
- George E. Collins and Alkiviadis G. Akritas. "Polynomial Real Root Isolation Using Descarte's Rule of Signs". In: SYMSAC. 1976, pp. 272–275. DOI: 10.1145/800205.806346.

References III

- Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Ábrahám. "SMT-RAT: An Open Source C++ Toolbox for Strategic and Parallel SMT Solving". In: SAT. Vol. 9340. 2015, pp. 360–368. DOI: 10.1007/978-3-319-24318-4_26.
- James H. Davenport and Joos Heintz. "Real Quantifier Elimination is Doubly Exponential". In: Journal of Symbolic Computation 5 (1–2 1988), pp. 29–35. DOI: 10.1016/S0747-7171(88)80004-X.
- Lloyd L. Dines. "Systems of Linear Inequalities". In: Annals of Mathematics 20.3 (1919), pp. 191–199. DOI: 10.2307/1967869.
- Bruno Dutertre and Leonardo de Moura. "A Fast Linear-Arithmetic Solver for DPLL(T)". In: CAV. 2006, pp. 81–94. DOI: 10.1007/11817963_11.
- Matthew England, Russell Bradford, and James H. Davenport. "Improving the Use of Equational Constraints in Cylindrical Algebraic Decomposition". In: ISSAC. 2015, pp. 165–172. DOI: 10.1145/2755996.2756678.
- Matthew England, Russell Bradford, James H. Davenport, and David Wilson. "Choosing a Variable Ordering for Truth-Table Invariant Cylindrical Algebraic Decomposition by Incremental Triangular Decomposition". In: ICMS. Vol. 8592. 2014. DOI: 10.1007/978-3-662-44199-2_68.
- Jean Baptiste Joseph Fourier. "Sur le Calcul des conditions d'inégalité". In: Nouveau Bulletin des Sciences par la Société philomathique de Paris (1825), pp. 66–68. URL: https://biodiversitylibrary.org/page/4153058.

References IV

- Jean Baptiste Joseph Fourier. “Solution d’une question particulière du calcul des inégalités”. In: Nouveau Bulletin des Sciences par la Société philomathique de Paris (1826), pp. 99–100. URL: https://biodiversitylibrary.org/page/4453516.
- Rebecca Haehn. “Using Equational Constraints in an Incremental CAD Projection”. Master’s thesis. RWTH Aachen University, 2018.
- Hoon Hong. “An Improvement of the Projection Operator in Cylindrical Algebraic Decomposition”. In: ISSAC. 1990, pp. 261–264. DOI: 10.1145/96877.96943.
- Zongyan Huang, Matthew England, David Wilson, James H. Davenport, Lawrence C. Paulson, and James Bridge. “Applying Machine Learning to the Problem of Choosing a Heuristic to Select the Variable Ordering for Cylindrical Algebraic Decomposition”. In: CICM. 2014, pp. 92–107. DOI: 10.1007/978-3-319-08434-3_8.
- Maximilian Jaroschek, Pablo Federico Dobal, and Pascal Fontaine. “Adapting Real Quantifier Elimination Methods for Conflict Set Computation”. In: FroCoS. Vol. 9322. 2015, pp. 151–166. DOI: 10.1007/978-3-319-24246-0_10. arXiv: 1511.01123.
- Dejan Jovanovic and Bruno Dutertre. “LibPoly: A Library for Reasoning about Polynomials”. In: SMT. CAV. Vol. 1889. 2017. URL: http://ceur-ws.org/Vol-1889/paper3.pdf.
- Dejan Jovanovi and Leonardo de Moura. “Solving Non-linear Arithmetic”. In: IJCAR. Vol. 7364. 2012, pp. 339–354. DOI: 10.1007/978-3-642-31365-3_27.

References V

- Marek Kota and Thomas Sturm. “A Generalized Framework for Virtual Substitution”. In: arXiv e-prints (2015). arXiv: 1501.05826.
- Gereon Kremer. “Cylindrical Algebraic Decomposition for Nonlinear Arithmetic Problems”. PhD thesis. RWTH Aachen University, 2020. URL: http://aib.informatik.rwth-aachen.de/2020/2020-04.pdf.
- Gereon Kremer and Erika Ábrahám. “Fully Incremental Cylindrical Algebraic Decomposition”. In: Journal of Symbolic Computation 100 (2020), pp. 11–37. DOI: 10.1016/j.jsc.2019.07.018.
- Gereon Kremer and Jens Brandt. “Implementing arithmetic over algebraic numbers A tutorial for Lazard’s lifting scheme in CAD”. In: SYNASC. 2021, pp. 4–10. DOI: 10.1109/SYNASC54541.2021.00013.
- Daniel Lazard. “An Improved Projection for Cylindrical Algebraic Decomposition”. In: Algebraic Geometry and its Applications. 1994. Chap. 29, pp. 467–476. DOI: 10.1007/978-1-4612-2628-4_29.
- Ulrich Loup, Karsten Scheibler, Florian Corzilius, Erika Ábrahám, and Bernd Becker. “A Symbiosis of Interval Constraint Propagation and Cylindrical Algebraic Decomposition”. In: CADE-24. Vol. 7898. 2013, pp. 193–207. DOI: 10.1007/978-3-642-38574-2_13.
- Scott McCallum. “An Improved Projection Operation for Cylindrical Algebraic Decomposition”. PhD thesis. University of Wisconsin-Madison, 1984. URL: https://research.cs.wisc.edu/techreports/1985/TR578.pdf.

References VI

- Scott McCallum. “On Projection in CAD-based Quantifier Elimination with Equational Constraint”. In: ISSAC. 1999, pp. 145–149. DOI: 10.1145/309831.309892.
- Scott McCallum and Hoon Hong. “On using Lazard’s projection in CAD construction”. In: Journal of Symbolic Computation 72 (2016), pp. 65–81. DOI: 10.1016/j.jsc.2015.02.001.
- Scott McCallum, Adam Parusiski, and Laurentiu Paunescu. “Validity proof of Lazard’s method for CAD construction”. In: Journal of Symbolic Computation 92 (2019), pp. 52–69. DOI: 10.1016/j.jsc.2017.12.002.
- Theodor Samuel Motzkin. “Beiträge zur Theorie der Linearen Ungleichungen”. PhD thesis. Universität Basel, 1936.
- Leonardo de Moura and Nikolaj Bjørner. “Proofs and Refutations, and Z3”. In: LPAR Workshops. Vol. 418. 2008, pp. 123–132. URL: http://ceur-ws.org/Vol-418/paper10.pdf.
- Leonardo de Moura and Dejan Jovanovi. “A Model-Constructing Satisfiability Calculus”. In: VMCAI. Vol. 7737. 2013, pp. 1–12. DOI: 10.1007/978-3-642-35873-9_1.
- Jasper Nalbach, Gereon Kremer, and Erika Ábrahám. “On Variable Orderings in MCSAT for Non-linear Real Arithmetic (extended abstract)”. In: SC^2. SIAM AG. Vol. 2460. 2019. URL: http://ceur-ws.org/Vol-2460/paper5.pdf.

References VII

- Volker Weispfenning. “Quantifier Elimination for Real Algebra — the Quadratic Case and Beyond”. In: Applicable Algebra in Engineering, Communication and Computing 8 (2 1997), pp. 85–101. DOI: 10.1007/s002000050055.
- Steven A. Wolfman and Daniel S. Weld. “The LPSAT Engine & its Application to Resource Planning”. In: IJCAI. 1999, pp. 310–316. URL: https://ijcai.org/Proceedings/99-1/Papers/046.pdf.