Teaching Mathematics and Statistics in Sciences - IPA HU-SRB/0901/221/088
Interesting Mathematical Problems in Sciences and Everyday Life - 2011

Effective Real Quantifier Elimination

Róbert Vajda

Bolyai Institute University of Szeged, Hungary
Aradi Vértanúk tere 1, Szeged, Hungary, 6720
vajdar@.u-szeged.hu
www.math.u-szeged.hu

In mathematical models, we often search for objects whose components satisfy certain constraints. Among the constraint satisfaction problems the simplest ones are the systems containing only equations and inequalities as constraints among the unknowns. We can be interested for satisfiability, i.e., for the existence of solutions, which might be dependent on certain parameters or we can search for the constructive description of the entire solution set. In this article we consider general algebraic systems over the reals.

It turns out that systems containing only algebraic equations and inequalities can be treated completely algorithmically and the constructive characterization of the solution sets are possible. The latter description is equivalent to an elimination of quantifiers from a logical formula which represents the system. The effective construction of the solution set is only tractable even for problems with modest input size with the best hardware and software environments. After the introduction of the problem (the elimination of quantifiers) and the algorithm for its solution (cylindrical algebraic decomposition), we illustrate some field of applications.

1. Introduction

Let us start with the first-order language of (the) elementary of algebra, that is, we start with a language in which where we can express the sum and the product of two elements, and additionally we can speak about the ordering of the elements. To do so, we give the signature of the formal language L_{or}[ordered rings]: $L_{\text{or}} = \{+,\ *,\ -,\ 0,\ 1; <\}$, where + and * are binary function symbols, - is a unary function symbol, 0 and 1 are constant symbols and < is a binary predicate symbol.

Following the usual inductive buildup of a logical language, we can define the terms, formulae and sentences of the the language L_{or}. E.g. $(1+1)*(1+0)$ is a term of the language, $0<1$ is an atomic formula of the language without variables, $\underset{x}{\forall}\underset{y}{\exists}(x+y=x)$ is a sentence, $\underset{y}{\exists}(x*x+y*y-4<0 \bigwedge y*y-2*x+2<0)$ is a formula of the language in which one free variable occurs.

We call any set of sentences a theory. If T is a theory (which usually consists of infinitely many formulae), $\Delta \subseteq T$, and T is the set of all consequences of Δ, then we say that Δ is an axiom system for T. We call “ordered fields” all those structures, which satisfy the axioms of ordering, the field axioms, and the axioms of monotonicity (they are the models of the theory of ordered fields). Special ordered fields are the real closed fields. These structures satisfy further axioms (axiom schemes), namely (1)-(3), the intuitive meaning of those additional axiom schemes is as follows: (1) All polynomials with odd degree have a root, (2) -1 never equals to sum of some squares, (3) all positive elements have a square root. The standard model for the theory of real closed fields (RCF) is the field of real numbers ℝ with the usual operations and ordering (more on the models of formal first-order theories and a slightly different axiomatization of RCF can be found in [8], and [13]).

(1) For all n≥0: $\underset{x_0,\ldots,x_{2n}}{\forall}\ \underset{y}{\exists}\ y^{2n+1}+\sum_{i=0}^{2n} x_i\, y^i = 0$

(2) For all n≥1: $\underset{x_1,\ldots,x_n}{\forall}\ x_1{}^2 + \ldots + x_n{}^2 + 1 \neq 0$

(3) $\underset{x}{\forall}\,\underset{y}{\exists}\left(\left(y^2 = x\right) \bigvee \left(y^2 + x = 0\right)\right)$

The main result for the RCF theory, first proven by Alfred Tarski in the thirties is, that this theory admits quantifier elimination, i.e. for any (arbitrarily quantified) formula ϕ of the language L_{or}there exists a formula ψ with the following properties:

(1) $\phi \Longleftrightarrow \psi$ [RCF $\vDash \phi \Leftrightarrow \psi$]

(2) ψ is quantifier free

(3) the free variables of ψ contained in the set of the free variables of ϕ.

Remarks.
1. Exploiting the existence of a Prenex form and the logical equivalences $\underset{x}{\forall}\phi \equiv \neg\underset{x}{\exists}\neg\phi$and $\underset{x}{\exists}(\phi \bigvee \psi) \equiv \left(\underset{x}{\exists}\phi \bigvee \underset{x}{\exists}\psi\right)$, it is not hard to see that it is enough to prove the quantifier elimination theorem for those formulae which have the following particular form: $\underset{x}{\exists}\phi$, where ϕ is a quantifier free conjunction of the atomic formula P ρ 0, where each P is a polynomial expression of some variables and $\rho\in\{<,=\}$. The proof of the theorem is still not obvious in this new setting.
2. If the original input formula ϕ does not contain free variables, then neither does the existing equivalent quantifier free formula ψ, consequently ψ is a ground formula of the language; for ground formulae it is always decidable whether they are true or not.
3. As a consequence of the point above (2), we get the strong result that the theory of RCF is decidable.

We can prove the above quantifier elimination theorem in essentially two different ways. Either we try to characterize with model-theoretic machinery the class of theories which admits quantifier elimination and then prove additionally that RCF belongs to this class. This approach is really fruitful; but the abstract model theoretic proof of the quantifier-elimination property does not give us a method which explicitly tells us how to carry out the elimination of the quantified variables. So we do not follow it in this introductory article further, since our motivation is the practical application of computers for doing and teaching mathematics. A second approach would be to give an explicit general algorithm for quantifier elimination and to prove the correctness of the algorithm. In fact this path was followed already by Tarski in his original paper [12], however a real algorithmic breakthrough came with G. E. Collins' method in the seventies [5], since the complexity of Collins' algorithm was much better than Tarski's. Collins' algorithm was first implemented in the QEPCAD program [QEPCAD B, Collins, Brown et al.], which is now freely available in the internet [2]. An improved variant of the algorithm can be currently found in the standard packages of some computer algebra systems [e.g.. *Mathematica* 5.0-; Strzebonski or Reduce-Redlog; Weispfenning]. Due to the author's experience, the implementation availabe in *Mathematica* has the best parameters for solving practical problems, for the average user. Besides the good computational performance, a further advantage is that the *Mathematica* computer algebra system has a flexible, nice front-end, so typing-in the problems and interpreting the outputs is relatively easy.

Now let us connect the above description of the quantifier elimination with the constraint systems that were mentioned in the abstract. We start with a relatively simple example. Take a system consisting of two linear equations with integer coefficients, as

```
2 x + y == 1 ∧ x - y == 2.
```

If one is interested in the solvability of the system, then we simply quantify existentially both variables and consider the input formula

$$\underset{x}{\exists}\ \underset{y}{\exists}\ (2\,x + y == 1 \wedge x - y == 2).$$

After the elimination of the quantified variables we get True (or 0==0), thus we decided (proved) the satisfiability of the system. Taking rather universal as existential quantifiers, we can prove identities and inequalities as well. Consider the well known inequality between different means:

$$\underset{x}{\forall}\ \underset{y}{\forall}\ \left(\left(x^2 + y^2\right) \big/ 2 \geq \left((x + y)^2 \big/ 4\right)\right).$$

If a theory is decidable, we may become suspicious, that only very few interesting mathematical problems are expressible in the frame of this formal theory and thus solvable automatically. We would like to demonstrate, with some introductory examples, that this is not the case for RCF. The examples were tested using *Mathematica* and QEPCAD. After the examples in which we only apply the quantifier elimination method as a black-box algorithm, we briefly describe the 'effective' Collins' quantifier elimination algorithm and mention its characteristic complexity properties.

2. Two introductory examples

Real roots of quadratics

First, let us characterize using the quantifier elimination method which polynomial with degree two and real coefficients has a real root. E.g. let our input formula be

$$\underset{x}{\exists}\ x\,{}^\wedge 2 + p\,x + q = 0.$$

As an output we gain the following necessary and sufficient condition for the free variables of the formula:

$$p^2 - 4\,q \geq 0.$$

```
Resolve[∃x (x^2 + p x + q == 0), Reals]
-p² + 4 q ≤ 0
```

REMARK 1

Mathematica 5.2-8.0: Usuage of the Resolve command.The name of the command which calls the quantifier elimination algorithm is "**Resolve**", we hope that via the examples the input and output syntax is clear.

Of course, we can also investigate with the method, when does a (formal) quadratic have two different real roots:

$$\underset{x_1,x_2}{\exists}\left(a\,x_1{}^2 + b\,x_1 + c = 0 \bigwedge a\,x_2{}^2 + b\,x_2 + c = 0 \wedge x_1 \neq x_2\right) \longrightarrow\longrightarrow\longrightarrow$$
$$\left(\left(a \neq 0 \bigwedge\ b^2 - 4\,a\,c > 0\right) \bigvee\ (a = 0 \wedge b = 0 \wedge c = 0)\right)$$

```
Resolve[∃{x1,x2} (a x1^2 + b x1 + c == 0 ∧ a x2^2 + b x2 + c == 0 ∧ x1 ≠ x2), Reals]
(a == 0 && b == 0 && c == 0) || (a ≠ 0 && -b^2 + 4 a c < 0)
```

Invertibility of a square matrix

Let $A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ be a 2×2 real matrix. Let us investigate when A isinvertible.

Using the definition of invertibility, A is invertible if and only if there exists an $X=\begin{pmatrix} x & y \\ u & v \end{pmatrix}$, such that $X\,A=I_2$; reformulating this condition with the matrix entries, we get:

$$\underset{x,y,u,v}{\exists}\ (x\,a+y\,c=1\wedge x\,b+y\,d=0\wedge u\,a+v\,c=0\wedge u\,b+v\,d=1)$$

As a solution of this quantifier elimination problem we get the quantifier free formula

$$a\,d-b\,c\neq 0.\ (+)$$

Of course we could also have gotten a slightly different solution formula, since a general algorithm may not know, which is the 'nicest' among the possibly infinitely many defining formula. But if the solution formula is $\phi_1(a,b,c,d)$and $\phi_2(a,b,c,d)$ is another quantifier free defining formula, then the sentence $\underset{a,b,c,d}{\forall}\ \phi_1 \Longleftrightarrow \phi_2$ is a true sentence of the theory. Additionally we notice, that QEPCAD gives in fact the form (+), while *Mathematica* doesn't.

$$a\,d-b\,c\neq 0$$

```
Resolve[
 Exists[{x, y, u, v}, x a + y c == 1 ∧ x b + y d == 0 ∧ u a + v c == 0 ∧ u b + v d == 1], Reals]
(a == 0 && b ≠ 0 && c ≠ 0) || (a ≠ 0 && -b c + a d ≠ 0)
```

3. Solotareff approximation problem

This is a polynomial approximation problem, where we approximate a real polynomial of degree n with polynomials having degree (at most) (n-2) in the closed interval [-1,1] using the uniform (supremum) norm. The natural question which arises in this setup, whether there exists a best approximation polynomial which satisfies the given conditions. It is maybe astonishing at the very first glance, that the problem can be expressed using the language L_{or}[ordered rings] and thus it is tractable with the quantifier elimination metodology.

To show this, let us begin the following symbolization of the problem:

$$\underset{\bar{Q}}{\forall}\ \underset{x}{\forall}\ (-1\leq x\leq 1)\Longrightarrow \|P-Q\|\leq \left\|P-\bar{Q}\right\|,$$

where P is given polynomial to approximate, Q is the best polynomial which satisfies the approxaimation conditions and $\bar{Q}$ ranges over the set of all polynomials with degree (n-2). The supremum norm (||.||) is obviously not part of the signature of the language L_{or} and we are not allowed to quantify in this first order language over polynomials, but it is easy to see, that the condition ||A||≤||B|| is equivalent to the following:

$$\underset{x}{\forall}\ (-1 \le x \le 1) \Longrightarrow \underset{y}{\exists}\ (-1 \le y \le 1 \wedge |A[x]| \le |B[y]|)$$

Furthermore, eliminating the absolute value (e.g. $|x| \le |y| \Longleftrightarrow x^2 \le y^2$) and introducing $3\,n - 1$ fresh variables [$(n+1)$ bound and $2\,(n-1)$ free variables for representing the coefficients of P, Q and $\overline{Q}$ resp.], we gain the sought for formalization of the problem. Another simple thought lead us to the conclusion that for arbitrary *n*, it is sufficient to approximate the following one parameter polynomial familiy of the form: $x^n + r\,x^{n-1}$ $(r \ge 0)$ (dear reader, please convience yourself).

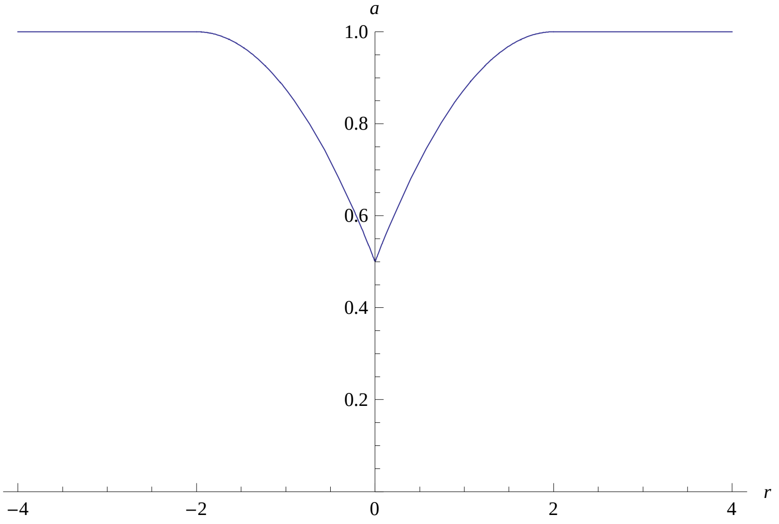
So let us formulate the quantifier elimiation problem which belongs to the Solotareff approximation problem for $n = 2$:

```
Timing[
 Resolve[r ≥ 0 ⋀ ∀{x,a1} (-1 ≤ x ≤ 1 ⇒ ∃y (-1 ≤ y ≤ 1 ⋀ (x² + r x - a)² ≤ (y² + r y - a1)²)),
  {a}, Reals]]
```

```
{19.2172, (0 ≤ r ≤ 2 && a == 1/8 (4 + 4 r - r²)) || (r > 2 && a == 1)}
```

```
              ⎧ 1                    r < -2
              ⎪ 1/2 - 1/2 r - r²/8   -2 ≤ r < 0
Sol22[r_] :=  ⎨ 1/2 + 1/2 r - r²/8   0 ≤ r ≤ 2
              ⎩ 1                    r > 2
```

```
Plot[Sol22[r], {r, -4, 4}, PlotRange → {0, 1},
 AxesOrigin → {0, 0}, AxesLabel → {r, a}]
```



The solution Q[r] is a continuous, piecewise polynomial function

Figure 1

Let us try to prove (verify) with elementary tools the result given by the algorithm.

Case 1: $r \geq 2$

In this case $P[x] = x\wedge 2 + r\,x$ is monoton increasing (its derivative is $2\,x + r$ and for all x∈[-1,1] $(2\,x + r) \geq -2 + r \geq 0$, thus P has a minimum in -1, a maximum in +1.
Conseqently the solution for a is the arithmetic mean of the function values in the left and the right endpoints of the interval, $\frac{P[-1]+P[1]}{2} = \frac{(1-r)+(1+r)}{2} = 1$.

Case 2: $0 \leq r \leq 2$

The maximum will be in +1 , but the the minimum is not in -1, since :
$P\,'[x] = 0 \Longleftrightarrow x = (-r/2)\,(\in [-1,\, 1])$ and a derivative changes sign at this point. The value at $-r/2$ is $-r^2/4$.
Thus in that case the aritmetic mean of the function values in 1 and in $-r/2$ is $\frac{-r^2}{8} + \frac{r}{2} + \frac{1}{2}$.

If we now increase the number of variables, which were involved in the second degree problem with 2 (one bound and one free), then we can investigate the approximation of cubic polynomials by the quantifier elimination method similarly. At the end of this section we treat only the special case from this probelm class, when $r = 0$, that is, when we approximate $P[x] = x^3$ with the linear polynomial family $\overline{Q}\,[x] = a_0\,x + b$. (Solution: a=3/4 and b=0, thus the sought for polynomial is $Q\,[x] = \frac{3}{4}\,x$)

In the next computation, we do not eliminate all quantifiers in one stroke, because the effective tractability of a QE-problem heavily depends on the order of the variables. First we eliminate the variables y, a_1 and $b_1$and in a second round we eliminate x.

In[2]:=

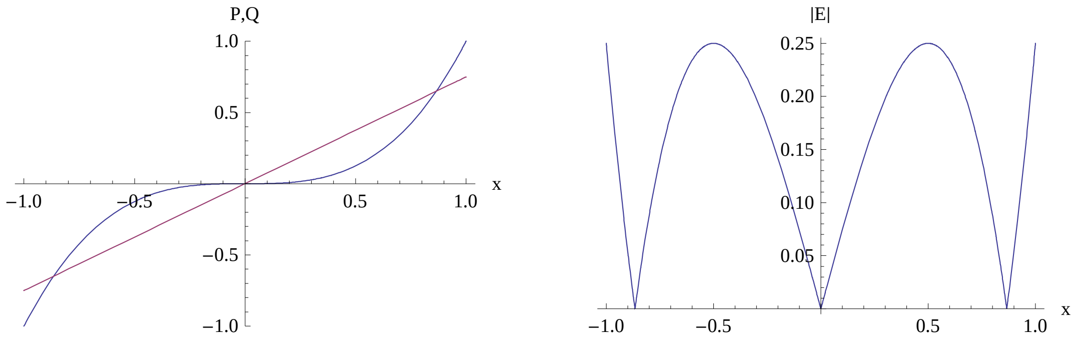
```
Timing[expr = Resolve[
   ∀{a1,b1} (-1 ≤ x ≤ 1 ⇒ ∃y (-1 ≤ y ≤ 1 && (x³ - a x - b)² ≤ (y³ - a1 y - b1)²)), Reals]]
```

```
{30.2939, x < -1 || (-1 ≤ x ≤ 1 && 1/4 (-1 - 4 a x + 4 x³) ≤ b ≤ 1/4 (1 - 4 a x + 4 x³)) || x > 1}
```

```
Timing[Resolve[∀{x} expr, Reals]]
```

```
{0.280017, a == 3/4 && b == 0}
```

```
Show[GraphicsArray[{Plot[{x^3, 3 / 4 x}, {x, -1, 1}, PlotRange → {-1, 1},
    DisplayFunction → Identity, AxesLabel → {"x", "P,Q"}],
   Plot[Abs[x^3 - 3 / 4 x], {x, -1, 1}, DisplayFunction → Identity,
    AxesLabel -> {"x", "|E|"}]}], ImageSize → {600, 300}]
```



a) The original cubic polynomial P and the best linear approximation Q. *b*) The graph of the error function $|E[x]| = |P[x] - Q[x]|$.

Figure 2

PROOF (Reasoning using elementary calculus)

It is easy to see that the optimal polynomial has a zero constant term. Therefore it is sufficient to consider the functions $\overline{Q}_a[x] = a\,x$.

Let E_a denote the function $P[x] - \overline{Q}_a[x]$. Since for all nonnegative a we have $|E_{-a}| \geq |E_a|$, we can assume that a is nonnegative. Furthermore, because of the central symmetry, it is sufficent to investigate the error function on [0,1]. We prove once more by case distinction.

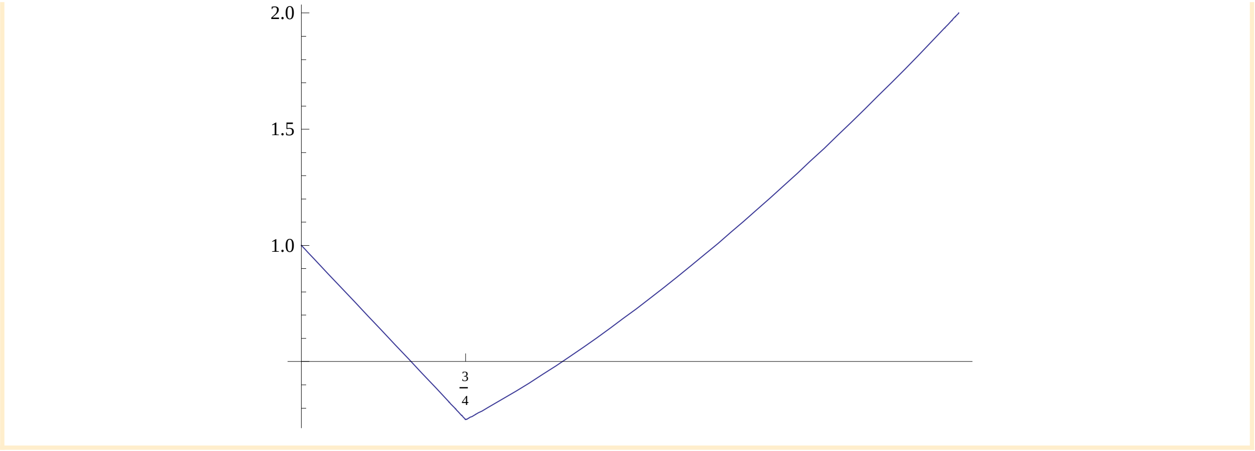
case 1: if $a \geq 3$ then $E_a[x]$ is monoton decreasing in [0,1], since $E_a'[x] = 3\,x^2 - a$ and thus for all x∈[0,1] and for all a∈(3,∞): $E_a'[x] \leq 3 - a \leq 0$
So the error equals to $|E[1]|$, which is $a - 1$; this is the smallest in case 1 if $a = 3$. Then the error is 2.

case 2: if $0 \leq a \leq 3$, then E has a local minimum on [0,1] at $(a/3)$^1/2, so the function to minimize in a has the form $\text{Max}[\text{f1}[a], \text{f2}[a]] = \text{Max}\left[\,|1-a|, \left(2/3\sqrt{3}\ a\text{^}(3/2)\right)\right]$, which has a global minimum at $a = 3/4$, and here the error is only 1/4, so the final solution is $Q[x] = \frac{3}{4}\,x$.

```
P = x^3 - a x;
f1 = P /. Solve[D[P, x] == 0, x][[2]]
```

$$-\frac{2\,a^{3/2}}{3\sqrt{3}}$$

```
Plot[Max[Abs[1 - a], -f1], {a, 0, 3}, Ticks → {{3 / 4}, Automatic}]
```



The investigated function has a global minimum at $a = \frac{3}{4}$.

Figure 3

We leave to the reader to investigate the parameter domain $r \geq 0$ in detail with the quantifier elimination or with other methods.

$$\underset{r,x,a_1,b_1}{\forall}\left((-1 \le x \le 1 \bigwedge 0 \le r \le 1) \Rightarrow \underset{y}{\exists}\left(-1 \le y \le 1 \bigwedge \left(x^3 + r\,x^2 - \left(\left(\frac{-1}{4} r^2 + \frac{1}{2} r + \frac{3}{4}\right) x + \left(\frac{-1}{108} r^3 + \frac{1}{6} r^2 + \frac{1}{4} r\right)\right)\right)^2 \le \left(y^3 + r\,y^2 - a_1\,y - b_1\right)^2\right)\right) \longrightarrow\longrightarrow\longrightarrow \text{True}$$

Finally we add as an interesting fact that due to the description given by E. Kaltofen [7], the Solotareff problem can be handled (with further mathematical background) at full scope with quantifier elimination until $n \leq 6$, while D. Lazard reported in 2005, that combining the quantifier elimination with other techniques the problem is solved, if $n \leq 10$.

■

4. The description of the algorithm working on the background and its properties

The basic idea of the Collins' real quantifier elimination algorithm is as follows: Without loss of generality we can assume that the arbitrarily quantified input formula ϕ is in prenex normal form and the right hand sides of the equailities and inequalities occuring in the quantifier free matrix of ϕ are reduced to 0. We extract all the multivariate polynomials from the formula and if r is the number of variables, then we decompose the r-space into disjunct connected subsets such that each extracted polynomial has a constant sign (positive, zero or negative) over each subset. This decomposition permits us to decide the truth conditions of the quantified input formula by using finitely many sample points (we pick a sample from each component of the decomposition). The algorithm is referred as Cylindrical Algebraic Decomposition (or with the abbreviation CAD) in the literature. The algorithm consists of three main phases: projection, base case (real root isolation and the decomposition of the real line) and lifting. The algorithm is recursive: in order to get the desired decomposition of $\mathbb{R}^r$, we need to construct decompositions of $\mathbb{R}^1$, $\mathbb{R}^2$, …, $\mathbb{R}^r$one after the other. We will see that exact symbolic computation is essential. The algorithm is doubly exponential in the number of variables r; fixing r, it is polynomial in the number of the involved polynomials and in the maximum degree of the involved polynoimals.

Basic notions

DEFINITION 1

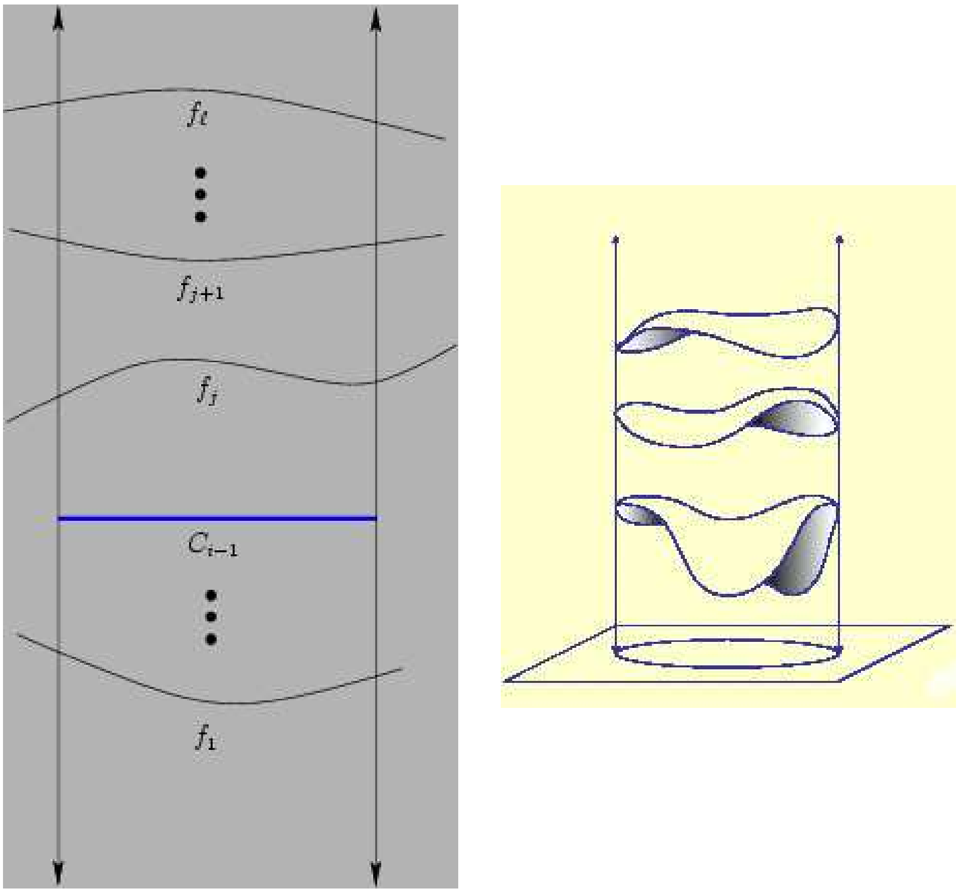
We call a nonempty connected subset of $\mathbb{R}^r$a region. A decomposition D of $X \subseteq \mathbb{R}^r$is finite disjunct collection of regions such that the union of the disjunct regions is X $\left(D = \{D_1 \ldots, D_\mu\};\ \bigcup D_i = X\right)$. We call an element of the decomposition a cell. The sample point of a cell is its arbitrary element.

DEFINITION 2

Let A be a finite set of polynomials (with integral coefficients). We say that that a decomposition is A-invariant, if all $a \in A$ has constant sign over each cell.

DEFINITION 3

Definition 3. A cylinder over a region R is $Z(R) = R \times \mathbb{R}$.



Stack construction with continuous functions in the 2- and 3-space.

Figure 4

Definition 4

Let $f_1, \ldots, f_k$ be real continuous functions defined on the region R. The decomposition of $Z(R)$ determined by the functions $f_1 \leq \ldots \leq f_k$ consists of (f_{i-1}, f_i)-sectors and f_i-segments. Such a decompositon is also called a stack determined by the $f_i^{'}\,s$ over the region R. (cf. Figure 4).

Definition 5

We say that the decomposition D of $\mathbb{R}^r$ is cylindrical, if either $r = 1$ and $D = \{D_1, \ldots, D_{2\nu+1}\}$, where $D_{2i} = \{\alpha_i\}$, $\alpha_i \in \mathbb{R}$, $\alpha_1 < \ldots < \alpha_\nu$ es $D_{2i+1} = (\alpha_i, \alpha_{i+1})$;
or $r > 1$ and there is a cylindrical decomposition $D' = \left\{D_1, \ldots, D_\mu\right\}$ of $\mathbb{R}^{r-1}$ such that $D = \left\{D_{1,1}, \ldots \mathrm{D}_{1,2\nu_1+1}, \ldots, D_\mu, \ldots, D_{\mu,2\nu_\mu+1}\right\}$, furthermore the decomposition $\left(D_{i,1} \ldots, D_{i,2\nu_i+1}\right)$ is a stack over D_i for all $1 \leq i \leq \mu$.

Definition 6

A decomposition of $\mathbb{R}^r$ is algebraic, if the connected sets in the decomposition are so called semialgebraic sets, that is, roughly speaking the functions f_i, which determine the decompositon, are algebraic [5, 13].

Definition 7

A decomposition of $\mathbb{R}^r$is a cylindrical algebraic decomposition, if it is cylindrical and algebraic.

Example 1: a univariate problem

Decide whether the following sentence ϕ of the theory is true or false.

$$\phi := \underset{x}{\forall}\,(x \geq 1) \Longrightarrow (x\text{^}2 + 3\,x > 2)$$

$A = \left\{x - 1,\, x^2 + 3\,x - 2\right\}$

roots: $\alpha_1 = -\frac{1}{2}\left(\sqrt{17} + 3\right)$, $\alpha_2 = -\frac{1}{2}\left(-\sqrt{17} + 3\right)$, $\alpha_3 = 1$; 7cells: 4 1-cells, 3 0-cells (red).
(No projection and lifting, only base case)

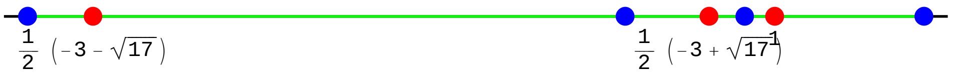
$D = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7\} = \{(-\infty, \alpha_1), \{\alpha_1\}, (\alpha_1, \alpha_2), \{\alpha_2\}, (\alpha_2, \alpha_3), \{\alpha_3\}, (\alpha_3, \infty)\}$

Sample points (red and blue points).

```
spoints = {-4, -1/2 (√17 + 3), 0, -1/2 (√17 - 3), 9/10, 1, 2};
```

```
Solve[x^2 + 3 x - 2 == 0, x]

{{x → 1/2 (-3 - √17)}, {x → 1/2 (-3 + √17)}}
```

```
N[%]

{{x → -3.56155}, {x → 0.561553}}
```



```
Map[(((x - 1) < 0 ∨ x² + 3 x - 2 > 0) /. x -> #) &, spoints]
{True, True, True, True, True, True, True}
```

Consequently, since the matrix evaluates over all cells to true; the quantifier free, to ϕ equivalent formula is True (i.e. ψ is 0=0).

```
Resolve[∀ₓ (x ≥ 1) ⇒ (x^2 + 3 x > 2), Reals]
True
```

Example 2: a bivariate problem

```
ϕ := ∃ (x^2 + y^2 ≤ 1 ∧ y > x)
     y
```

$A = \{-y^2 - x^2 + 1,\ y - x\}$

Phase I. In case of bivariate polynomials it is sufficient to include into the first projection set P_1 the resultants of each different polynomial pair and the resultant of each polynomial from A and its derivative (with respect to the main variable y) .

$\mathrm{Res}_y[-y^2 - x^2 + 1,\ -2\,y] = 4\,(1 - x^2)$
$\mathrm{Res}_y[-y^2 - x^2 + 1,\ y - x] = 1 - 2\,x^2$
$P_1 = \{1 - x^2,\ 1 - 2\,x^2\}$

In the general case the projection factor set P_1 is more complicated: we have to determine the principal subresultant coefficients of the polynomial pairs from the reducta set of A, but even then we obtain a finite set of $(r-1)$-variate polynomials.

```
A = {-y^2 - x^2 + 1, y - x};
```

```
Resultant[-y^2 - x^2 + 1, -2 y, y]
4 (1 - x^2)
```

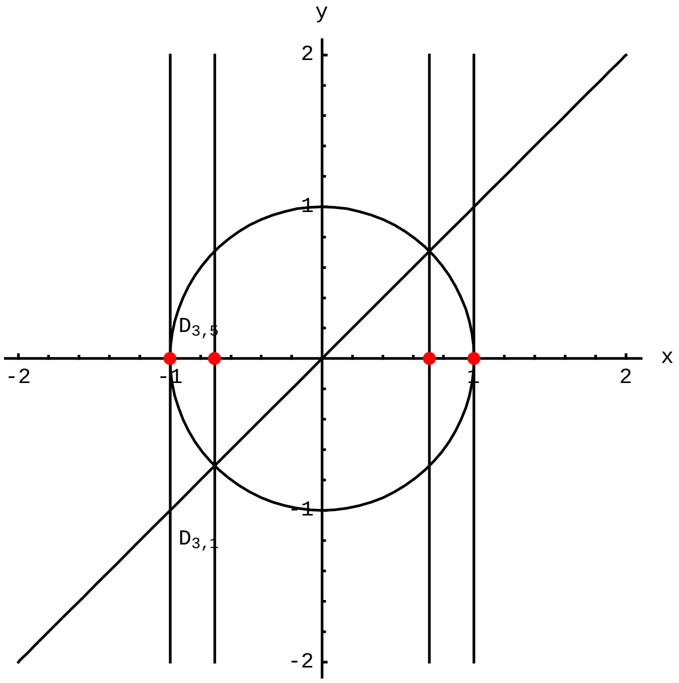
```
Resultant[y - x, -y^2 - x^2 + 1, y]
1 - 2 x^2
```

Phase II. The real root isolation of the polynomials in P_1 delivers the decomposition of the real line $\mathbb{R}^1$: there are 9 cells alltogether. As sample point we choose a rational number from the 1-cells and in case of the 0-cells we choose the point itself. We could notice, that the roots of the univariate polynomials in P_1 detect the 'singularities' (common points, self-crossings, tangential points, cusps, isolated points etc.)of the algebraic curves determined by the original bivariate polynomials in A.

$\alpha_1 = -1,\ \alpha_2 = -\frac{1}{\sqrt{2}},\ \alpha_3 = \frac{1}{\sqrt{2}},\ \alpha_4 = 1$

$$D' = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9\} =$$
$$\{(-\infty, \alpha_1), \{\alpha_1\}, (\alpha_1, \alpha_2), \{\alpha_2\}, (\alpha_2, \alpha_3), \{\alpha_3\}, (\alpha_3, \alpha_4), \{\alpha_4\}, (\alpha_4, \infty)\}$$

```
spoints1 = {-2, -1, -4 / 5, -1 / Sqrt[2], 0, 1 / Sqrt[2], 4 / 5, 1, 2};
```



Stack construction

Figure 5

Phase III. Stack construction over one dimensional cells, lifting of the decomposition of $\mathbb{R}$: We substitute back each sample point to the original bivariate polynomials in A. Doing this we get 9×2 univariate polynomials. Carrying over the root isolation of the pairs of polynomials, we get decompositions of the cylinders defined by the 0– and 1–cells. For instance, if we take the third pair $\{9/25 - y^2, 4/5 + y\}$, there exists three real roots, thus we get 7 cells; once more we take samples from the generated 2 dimensional cells.

$f_{3,1} = x \leq f_{3,2} = -\sqrt{1-x^2} \; \leq f_{3,3} = \sqrt{1-x^2}$ (over D_3)

$D_3 = \{D_{3,1}, D_{3,2}, D_{3,3}, D_{3,4}, D_{3,5}, D_{3,6}, D_{3,7}\} = \{(-\infty,\; f_{3,1}), \{f_{3,1}\}, (f_{3,1},\; f_{3,2}), \{f_{3,2}\}\,(f_{3,2},\; f_{3,3}), \{f_{3,3}\}, (f_{3,3},\; \infty)\}$

```
Map[(A /. x → #) &, spoints1]
```

$$\left\{\{-3-y^2,\ 2+y\},\ \{-y^2,\ 1+y\},\ \left\{\frac{9}{25}-y^2,\ \frac{4}{5}+y\right\},\ \left\{\frac{1}{2}-y^2,\ \frac{1}{\sqrt{2}}+y\right\},\ \{1-y^2,\ y\},\ \left\{\frac{1}{2}-y^2,\ -\frac{1}{\sqrt{2}}+y\right\},\ \left\{\frac{9}{25}-y^2,\ -\frac{4}{5}+y\right\},\ \{-y^2,\ -1+y\},\ \{-3-y^2,\ -2+y\}\right\}$$

```
Sort[Flatten[y /. Map[Solve[# == 0] &, %[[3]]]]]
```

$$\left\{-\frac{4}{5},\ -\frac{3}{5},\ \frac{3}{5}\right\}$$

```
spoints2 =
  {
   {{-2, -3}, {-2, -2}, {-2, 0}},
   {{-1, -2}, {-1, -1}, {-1, -1 / 2}, {-1, 0}, {-1, 1}},
   {{-4 / 5, -1}, {-4 / 5, -4 / 5}, {-4 / 5, -7 / 10},
    {-4 / 5, -3 / 5}, {-4 / 5, 0}, {-4 / 5, 3 / 5}, {-4 / 5, 1}},
   {{-1/ √2, -1}, {-1/ √2, -1/ √2}, {-1/ √2, 0},
    {-1/ √2, 1/ √2}, {-1/ √2, 1}},
   {{0, -2}, {0, -1}, {0, -1 / 2}, {0, 0}, {0, 1 / 2}, {0, 1}, {0, 2}},
   {{1/ √2, -1}, {1/ √2, -1/ √2},
    {1/ √2, 0}, {1/ √2, 1/ √2}, {1/ √2, 1}},
   {{4 / 5, -1}, {4 / 5, -3 / 5}, {4 / 5, 0}, {4 / 5, 3 / 5},
    {4 / 5, 7 / 10}, {4 / 5, 4 / 5}, {4 / 5, 1}},
   {{1, -1}, {1, 0}, {1, 1 / 2}, {1, 1}, {1, 2}},
   {{2, 0}, {2, 2}, {2, 3}}
  };
```

```
trv =
 Map[Map[((1 - x^2 - y^2 ≥ 0 ∧ y - x > 0) /. {x → #[[1]], y → #[[2]]}) &, #] &, spoints2]
```

```
{{False, False, False}, {False, False, False, True, False},
 {False, False, False, True, True, True, False}, {False, False, True, True, False},
 {False, False, False, False, True, True, False},
 {False, False, False, False, False},
 {False, False, False, False, False, False, False},
 {False, False, False, False, False}, {False, False, False}}
```

We extract the labels of the 'good' cells involved in the decomposition of ℝ and construct the final defining formula.

```
Union[Flatten[MapIndexed[Cases[#1, True → #2] &, %]]]
```

```
{2, 3, 4, 5}
```

```
Simplify[(x == -1) ⋁ (-1 < x < -1 / Sqrt[2]) ⋁
   (x == -1 / Sqrt[2]) ⋁ (-1 / Sqrt[2] < x < 1 / Sqrt[2])]
```

$$-1 \le x < \frac{1}{\sqrt{2}}$$

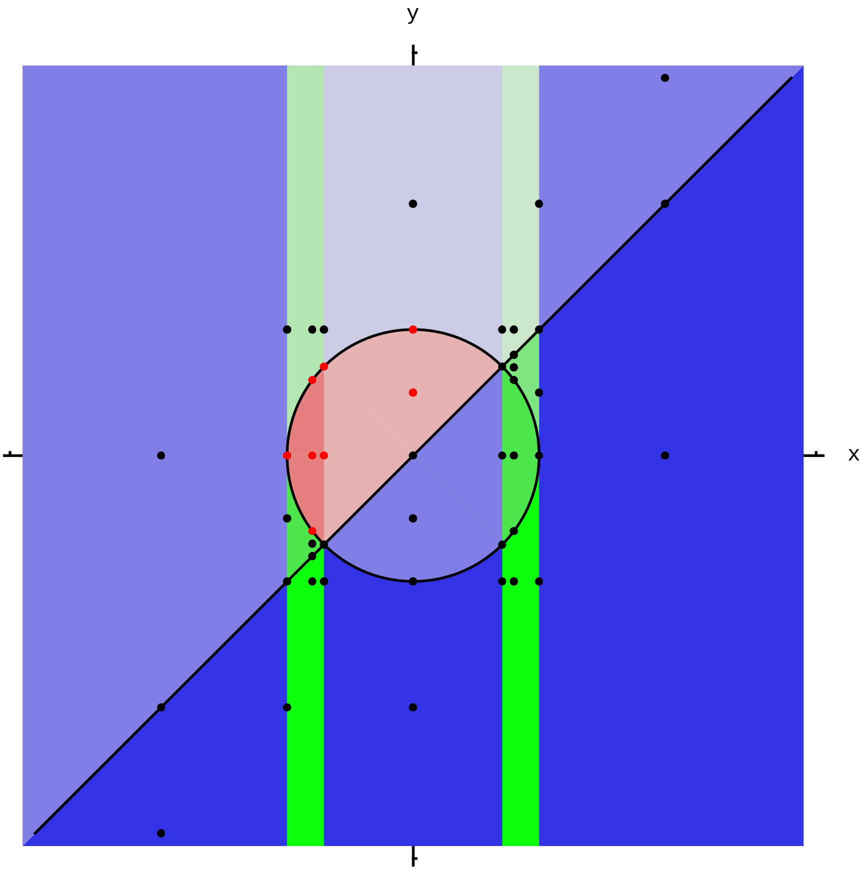
```
Resolve[∃y (x^2 + y^2 ≤ 1 ⋀ y > x), Reals]
```

$$-1 \le x < \frac{1}{\sqrt{2}}$$

Remark 2

In this examples in fact we gave the quantifier-free solution formula ψ in an extended language, but it can be shown, that the solution formula can be also defined using the signs of the irreducible factors of the polynomials in the augmented projection set; the combination of these sign-conditions is compatible with the signature of the original language, e.g.

$\psi: \left(1-2x^2>0\right) \bigvee \left(1-2x^2=0 \bigwedge \boldsymbol{x}<0\right) \bigvee (x+1=0) \bigvee \left(1-2x^2<0 \bigwedge \boldsymbol{x}<0 \bigwedge x+1>0\right)$



Cylindrical algebraic decomposition of $\mathbb{R}^2$; Those sample points and the cells of sample points are plotted in red, which satisfy the polynomial constraints `(x^2 + y^2 ≤ 1 ⋀ y > x)` in the matrix of the prenex formula. At total nine different stacks could be seen in the picture.

Figure 6

5. Conclusion

In this survey we considered some mathematical applications of the quantifier elimination method over the theory of real closed fields. We briefly described the core of the Collins algorithm, which solves the quantifier elimination problem by giving a sign invariant decomposition of the *r*-space. We did not touch those mathematical applications which would require more sophisticated background, nor the industrial applications like the piano-movers problem in robotics. We demonstrated through examples the main steps of the algorithms and gave its complexity. In the recent years several important improvements have been made to the original method. Obviously, the method could not only be used by itself, but rather as a component of a general purpose mathematical assistant system. For instance, B. Buchberger's PCS method [3, 10] successfully integrates the CAD method into an automated reasoning system which facilitates the reasoning in the field of elementary analysis.

References

[1] D. S. Arnon -G. E. Collins - S. McCallum: Cylindrical Algebraic Decomposition I: The Basic Algorithm. In: Caviness-Johnson (eds): Quantifier elimination and cylindrical algebraic decomposition (pp. 136-151), Springer 1998.

[2] C. W. Brown: An Overview of QEPCAD B: a Tool for Real Quantifier Elimination and Formula Simplification. Journal of Japan Society for Symbolic and Algebraic Computation, 10(1) pp. 13-22, 2003.

[3] B. Buchberger. The PCS Prover in Theorema. In: Lecture Notes in Computer Science (LNCS) 2178, pp. 19-23, Springer Verlag, Berlin, 2001.

[4] G. E. Collins: Quantifier Elimination by Cylindrical Algebraic Decomposition — Twenty years of progress. In: Caviness-Johnson (eds): Quantifier elimination and cylindrical algebraic decomposition (pp. 8-23), Springer 1998.

[5] G. E. Collins: Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In . In: Caviness-Johnson (eds): Quantifier elimination and cylindrical algebraic decomposition (pp. 85-121), Springer 1998.

[6] G. E. Collins: Application of Quantifier Elimination to Solotareff's Approximation Problem. Technical report no. 95-31 in RISC Report Series, University of Linz, Austria. 1995.

[7] E. Kaltofen: Challenges of Symbolic Computation: My Favorite Open Problems. In: Journal of Symbolic Computation (JSC), 29(6), pp. 891-919, 2000.

[8] D. Marker: Model Theory: An introduction. GTM 217, Springer 2002.

[9] H. Rolletchek: Decidable Logical Theories (Lecture notes), RISC Linz, 2003.

[10] A. Strzebonski: Cylindrical Algebraic Decomposition Using Validated Numerics. Paper presented at the ACA 2002 Session on Symbolic-Numerical Methods in Computational Science, Volos, Greece, 2002.

[11] A. Szendrei: Discrete Mathematics: Logic, Algebra and Combinatorics (in Hungarian) Polygon, Szeged 1994.

[12] A. Tarski: Decision Method for Elementary Algebra and Geometry In: Caviness-Johnson (eds): Quantifier elimination and cylindrical algebraic decomposition (pp. 24-84), Springer 1998.

[13] F. Winkler: Polynomial Algorithms in Computer Algebra, Chapter 9, Quantifier elimination in real closed fields. (pp. 204-214), Springer 1998.

Appendix. Dynamic Demonstrations

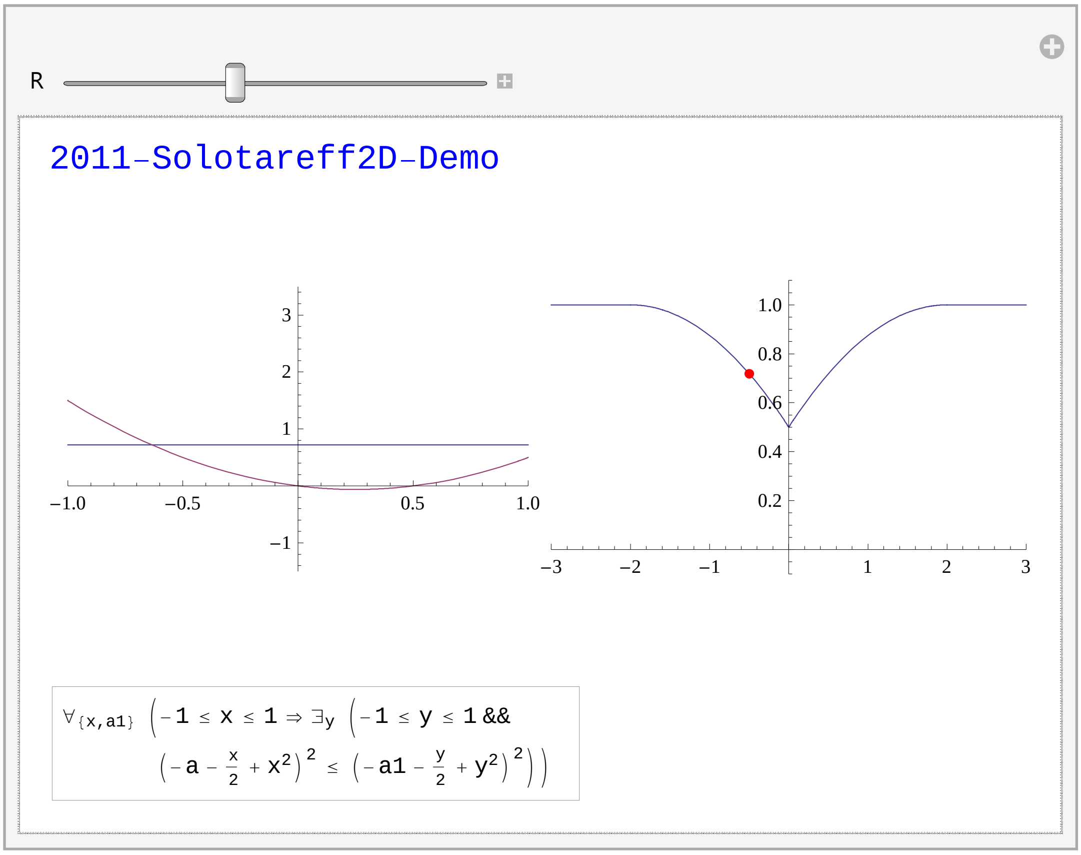
Solotareff Demo (n=2)

Description: Change the value of the parameter *R* by the slider. In each round you will see the quadratic polynomal (red), the best approximating constant polynomial (blue) and to the right a new plot with the graph of the optimal solutions. Moreover, the input formula of the quantifier elimination problem associated to the Solotareff problem is printed, too.

In[5]:=

```
Manipulate[Column[{Row[{Style["2011-Solotareff2D-Demo", Blue, FontSize → 16]}],

     Row[{Plot[{{ 1                  r < -2
                  1/2 - 1/2 r - r²/8  -2 ≤ r < 0
                  1/2 + 1/2 r - r²/8  0 ≤ r ≤ 2     /. r → R, x² + R x}, {x, -1, 1},
                  1                  r > 2

        PlotRange → {{-1, 1}, {-3/2, 7/2}}, ImageSize → {230, 230}],

         Plot[{ 1                  r < -2
                1/2 - 1/2 r - r²/8  -2 ≤ r < 0
                1/2 + 1/2 r - r²/8  0 ≤ r ≤ 2     , {r, -3, 3},
                1                  r > 2

        PlotRange → {{-3, 3}, {-.1, 1.1}}, ImageSize → {230, 230}, Epilog → {Red,

           PointSize[.02], Point[{R, { 1                  r < -2
                                       1/2 - 1/2 r - r²/8  -2 ≤ r < 0
                                       1/2 + 1/2 r - r²/8  0 ≤ r ≤ 2     /. r → R}]}]}],
                                       1                  r > 2

     InputField[∀{x,a1} (-1 ≤ x ≤ 1 ⇒ ∃y (-1 ≤ y ≤ 1 ⋀ (x² + R x - a)² ≤ (y² + R y - a1)²)),

      Enabled → False]}],

  {{R, -1/2}, -5/2, 5/2, 1/10}]
```

Out[5]=



$$\forall_{\{x,a1\}}\left(-1\le x\le 1\Rightarrow \exists_y\left(-1\le y\le 1\,\&\&\,\left(-a-\frac{x}{2}+x^2\right)^2\le\left(-a1-\frac{y}{2}+y^2\right)^2\right)\right)$$

1D CAD Demo

Description: Type a well-formed existential sentence with one bound variable to the inputfield. Be the formula matrix a conjunction of polynomial relations. Use only the "==, >, <" relations.

The demo generates a sign-invariant-decomposition of the real line, i. e., a samplepointlist with a sign matrix (-1, 0, +1).

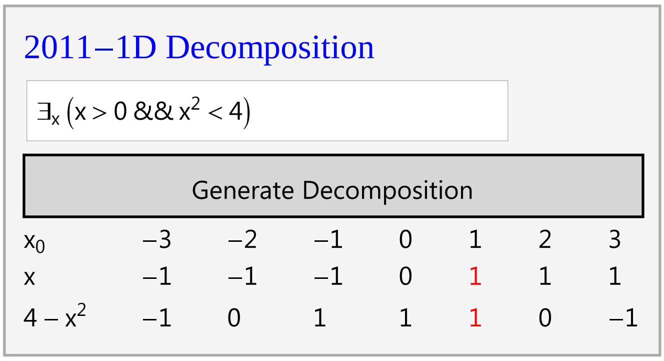
If the existential sequence is valid, you will see some red columns of the sign-matrix.

Whenever you modify the formulamatrix, click to the button to get the new sign-matrix.

In[1]:=

```
CAD1[y_] := Module[{RSignList, ty, plist, zerolist, samplelist}, trrules =
   {Greater[p_, q_] → p - q > 0, Equal[p_, q_] → p - q == 0, Less[p_, q_] → q - p > 0};
  ty = y /. trrules;
  RSignList = Map[If[(Head[#] === Greater), 1, 0] &,
    If[Head[ty[[2]]] === And, List @@ ty[[2]], {ty[[2]]}]];
  plist = Cases[ty, (Equal[p_, 0] | Greater[p_, 0]) → p, Infinity];
  zerolist = Union[Sort[DeleteCases[Join @@
       Map[x /. Solve[#] &, Thread[Equal[plist, 0]]], Complex[_, _]], Less]];
  samplelist = If[zerolist == {}, {0}, Sort[Join[Sort[Join[
         Table[(zerolist[[k]] + zerolist[[k + 1]]) / 2, {k, Length[zerolist] - 1}],
         zerolist], Less], {zerolist[[1]] - 1, Last[zerolist] + 1}], Less]];
  TableForm[Prepend[Transpose[Transpose[Table[
         Prepend[Table[plist[[j]] /. x → samplelist[[k]] // Simplify // Sign,
          {k, Length[samplelist]}], plist[[j]]], {j, Length[plist]}]] /.
      RSignList → Map[Style[#, Red] &, RSignList]], Join[{"x₀"}, samplelist]]]]
```

```
Panel[DynamicModule[{form = Exists[x, x > 0 ∧ x^2 < 4]},
  Column[{Text[Style["2011-1D Decomposition", Blue, FontSize → 16]], InputField[
    Dynamic[form]], Button["Generate Decomposition"], Dynamic[CAD1[form]]}]]]
```



2011–1D Decomposition

$\exists_x (x > 0 \&\& x^2 < 4)$

x_0	−3	−2	−1	0	1	2	3
x	−1	−1	−1	0	1	1	1
$4 - x^2$	−1	0	1	1	1	0	−1