Quantifier Elimination by Cylindrical Algebraic Decomposition Based on Regular Chains

Changbo Chen[1] and Marc Moreno Maza[2]
(Gratitude goes to James Davenport for presenting this talk)

[1] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences
[2] ORCCA, University of Western Ontario

July 23, 2014
ISSAC 2014, Kobe, Japan

Outline

Outline

Quantifier Elimination

- Input: a prenex formula $PF := (Q_{k+1}x_{k+1} \cdots Q_n x_n)\, F(x_1, \ldots, x_n)$
  - $F(x_1, \ldots, x_n)$: a quantifier free formula over $\mathbb{R}$
  - each Q_i is either $\exists$ or $\forall$.
- Output : a quantifier free formula $SF(x_1, \ldots, x_k)$ such that $SF \Leftrightarrow PF$ holds for all $x_1, \ldots, x_k \in \mathbb{R}$.

Quantifier Elimination (QE)

$(\exists x)(\forall y)\ (ax^2 + bx + c) - (ay^2 + by + c) \geq 0$, where $a, b, c, x, y \in \mathbb{R}$,

for which QE yields

$$(a < 0) \vee (a = b = 0).$$

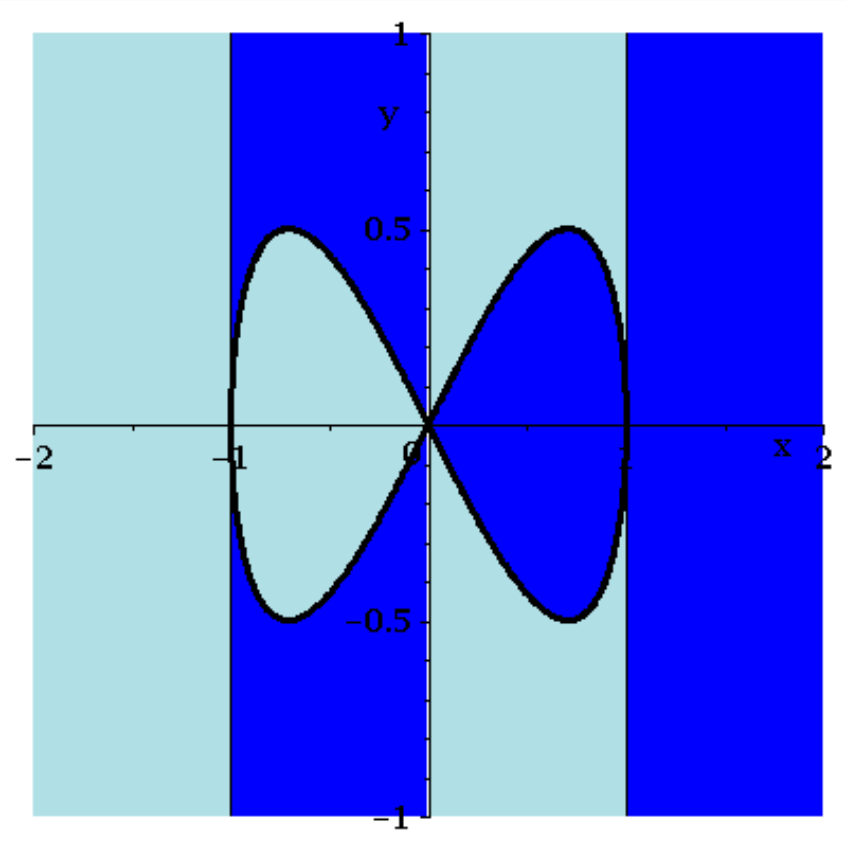
Quantifier Free Formula (QFF)

$$\neg(y - x^2 > 0 \wedge z^3 - x = 0) \vee (z + xy \geq 0 \wedge x^2 + y^3 \neq 0)$$

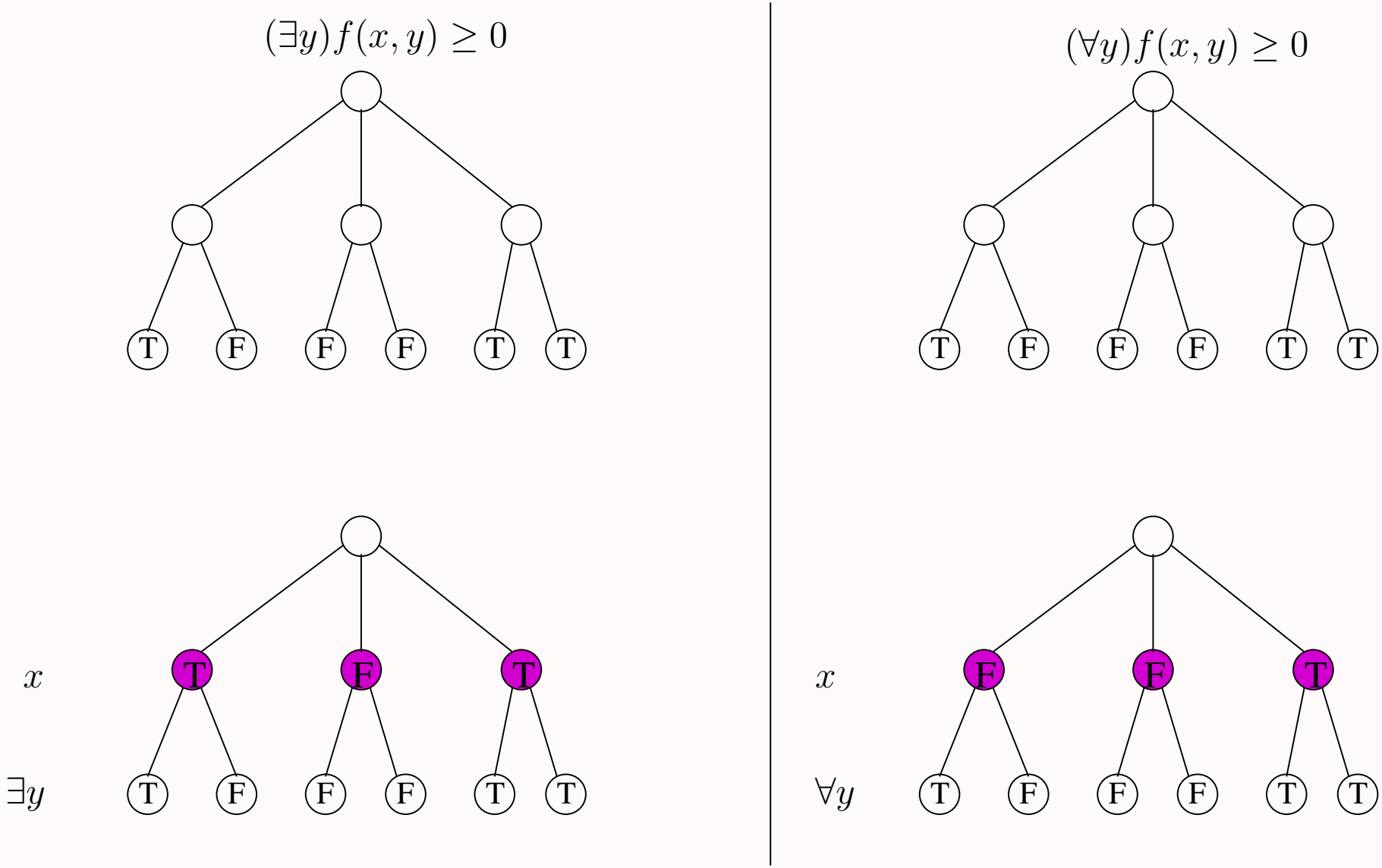
Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^n$

A CAD of $\mathbb{R}^n$ is a partition of $\mathbb{R}^n$ such that each cell in the partition is a connected semi-algebraic subset of $\mathbb{R}^n$ and all the cells are cylindrically arranged.

Two subsets A and B of $\mathbb{R}^n$ are called cylindrically arranged if for any $1 \leq k < n$, the projections of A and B on $\mathbb{R}^k$ are either equal or disjoint.



Why CAD supports QE : The main idea



CAD based on regular chains (RC-CAD)

Motivation: potential drawback of Collins' projection-lifting scheme

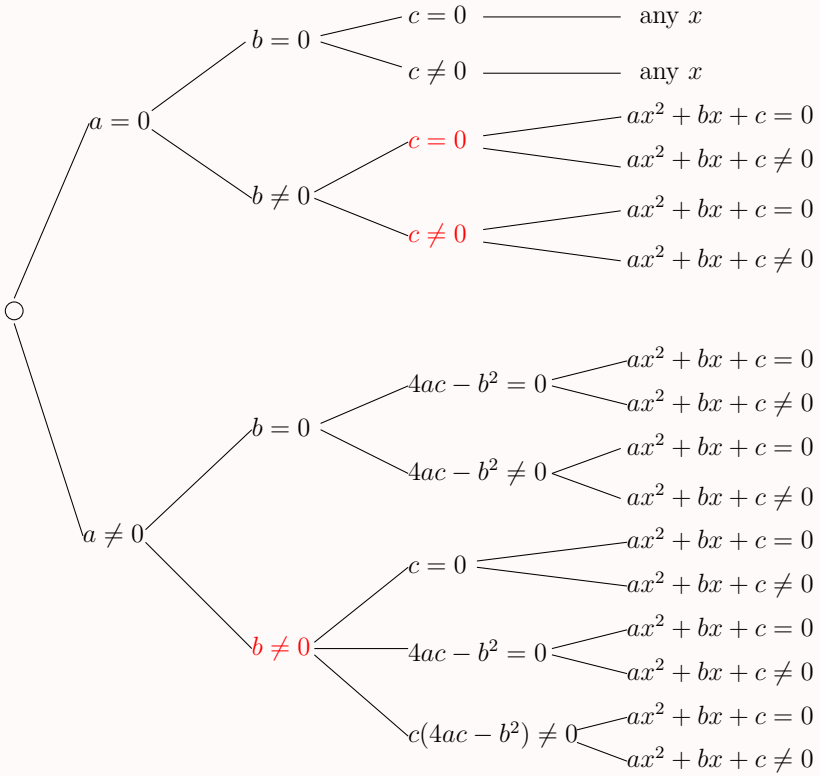
- The projection operator is a function defined independently of the input system.
- As a result, a strong projection operator (Collins-Hong operator) usually produces much more polynomials than needed.
- A weak projection operator (McCallum-Brown operator) may fail for non-generic cases.

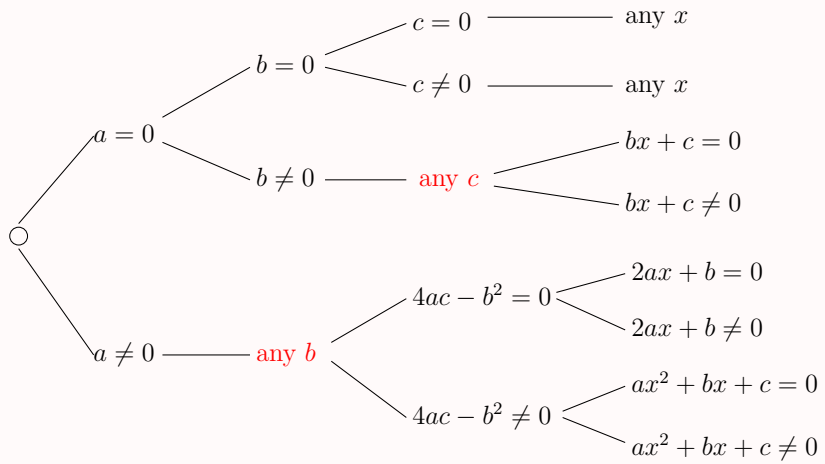
Solution: make case discussion during projection

- Case discussion is common for algorithms computing triangular decomposition.
- At ISSAC'09, we (with B. Xia and L. Yang) introduced case discussion into CAD computation.
- The new method consists of two phases. The first phase computes a complex cylindrical tree (CCT). The second phase decomposes each cell of CCT into its real connected components.

Illustrate PL-CAD and RC-CAD by parametric parabola example

Let $f := ax^2 + bx + c$. Suppose we'd like to compute f-sign invariant CAD. The projection factors are $a, b, c, 4ac - b^2, ax^2 + bx + c$. Let's rethink PL-CAD in terms of a *complex cylindrical tree*, see left tree.





Clearly, RC-CAD (see right tree) computes a smaller tree by *avoiding useless case distinction.*

QE by RC-CAD

Challenges for doing QE by RC-CAD

- RC-CAD has *no global projection factor set* associated to it.
- Instead, it is associated with a complex cylindrical tree. The polynomials in one path of a tree may not be sign invariant above cells derived from a different path of a tree.
- There is *no universal projection operator for RC-CAD.*
- Refining an existing CAD is not straightforward comparing to PL-CAD.

The solution

- Uses an operation introduced in ASCM 2012 (C. Chen, M. Moreno Maza) for *refining a complex cylindrical tree* and,
- Adapts C. W. Brown's incremental method for creating *projection-definable* PL-CAD to RC-CAD;
- The approach works with truth-invariant CAD produced in ASCM 2012 and CASC 2014 (with R. Bradford, J. H. Davenport, M. England and D. J. Wilson) for making use of equational constraints.

QE by RC-CAD : The big picture

Algorithm: QuantifierElimination

- Input: A prenex formula $PF := (Q_{k+1}x_{k+1} \cdots Q_n x_n) FF(x_1, \ldots, x_n)$.
- Ouput: A solution formula of PF.

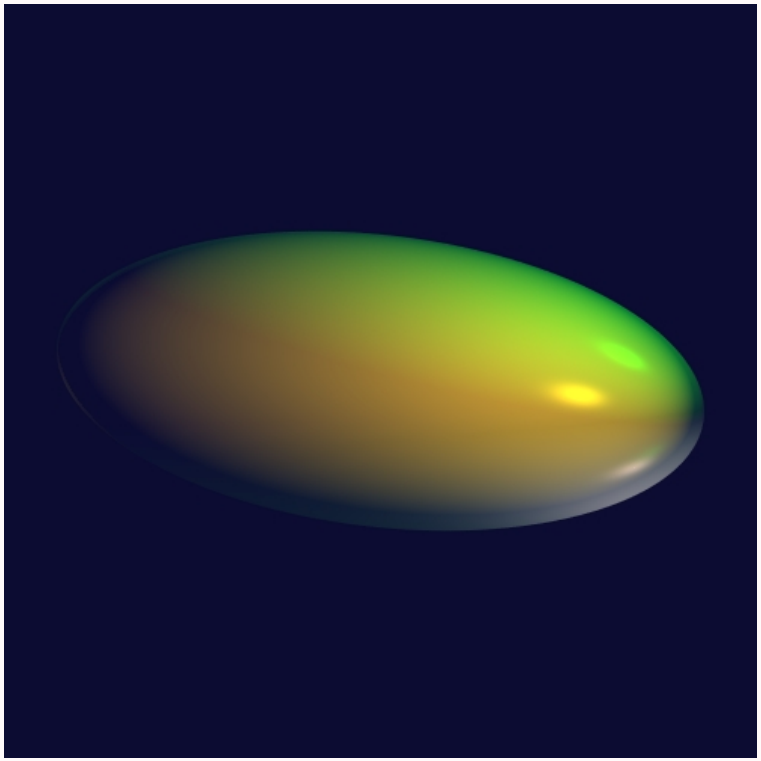
Description

1. Let F be the set of polynomials appearing in FF
2. $T :=$ CylindricalDecompose(F)//computes a complex cylindrical tree
3. $RT :=$ MakeSemiAlgebraic(T)//computes a CAD tree
4. AttachTruthValue(FF, RT)//evaluate the truth values of FF at each cell
5. PropagateTruthValue(PF, RT)//get the true values of PF
6. MakeProjectionDefinable(PF, RT)//refine RT until projection definable
7. $SF :=$ GenerateSolutionFormula$_k(RT)$//generate QFF describing true cells in free space

Outline

The first example

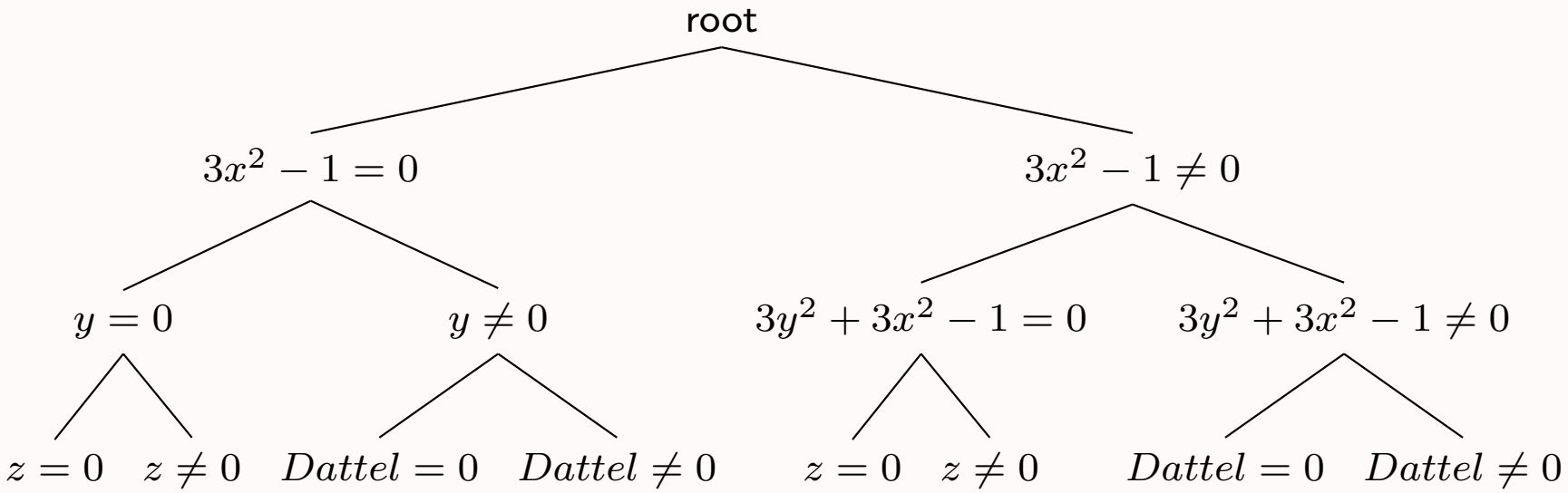
Let $Dattel := z^2 + 3y^2 + 3x^2 - 1$.



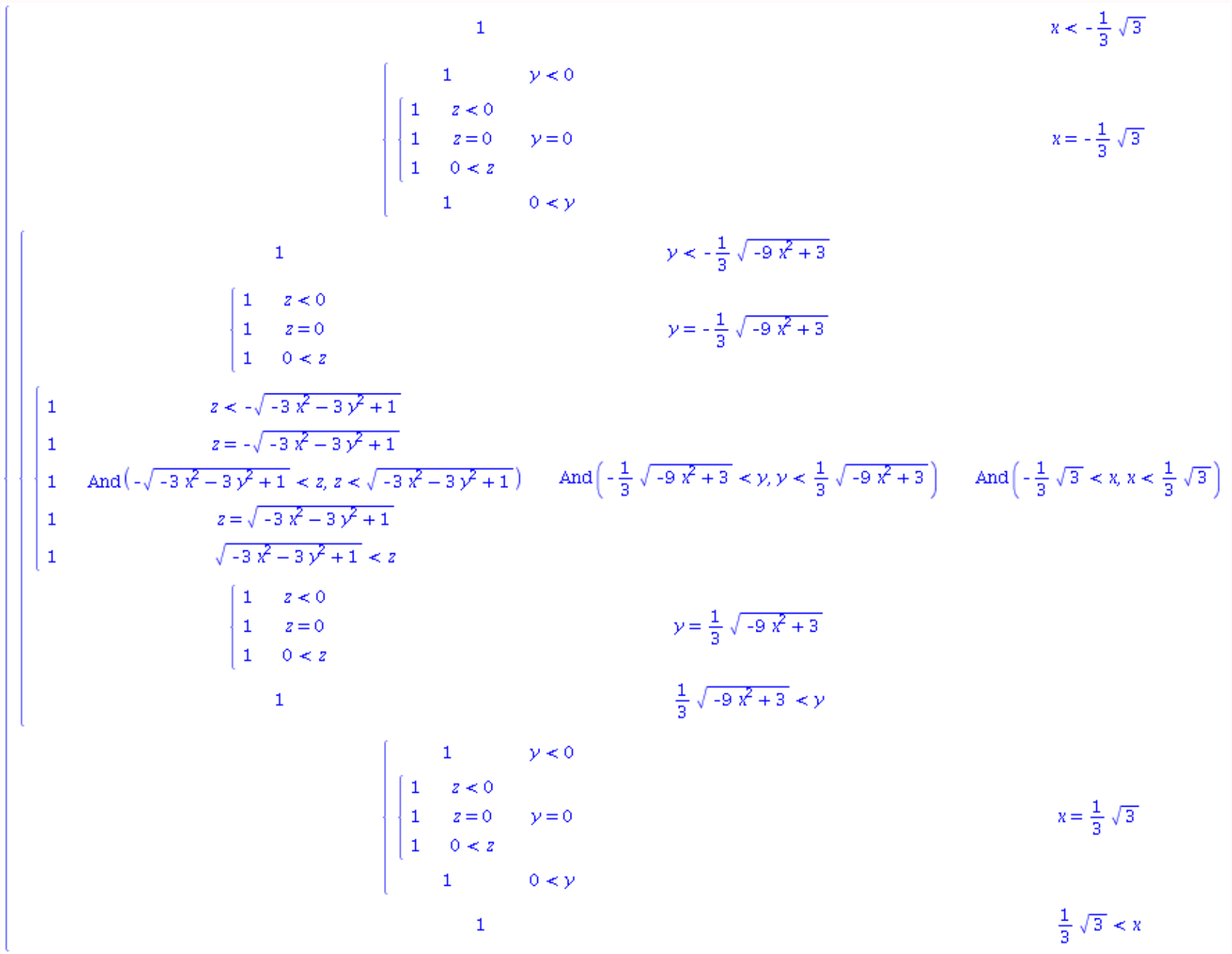
`http://www1-c703.uibk.ac.at/mathematik/project/bildergalerie/gallery/dattel.jpg`

Let $f := (\exists z)\ Dattel = 0$ be the input existential formula.

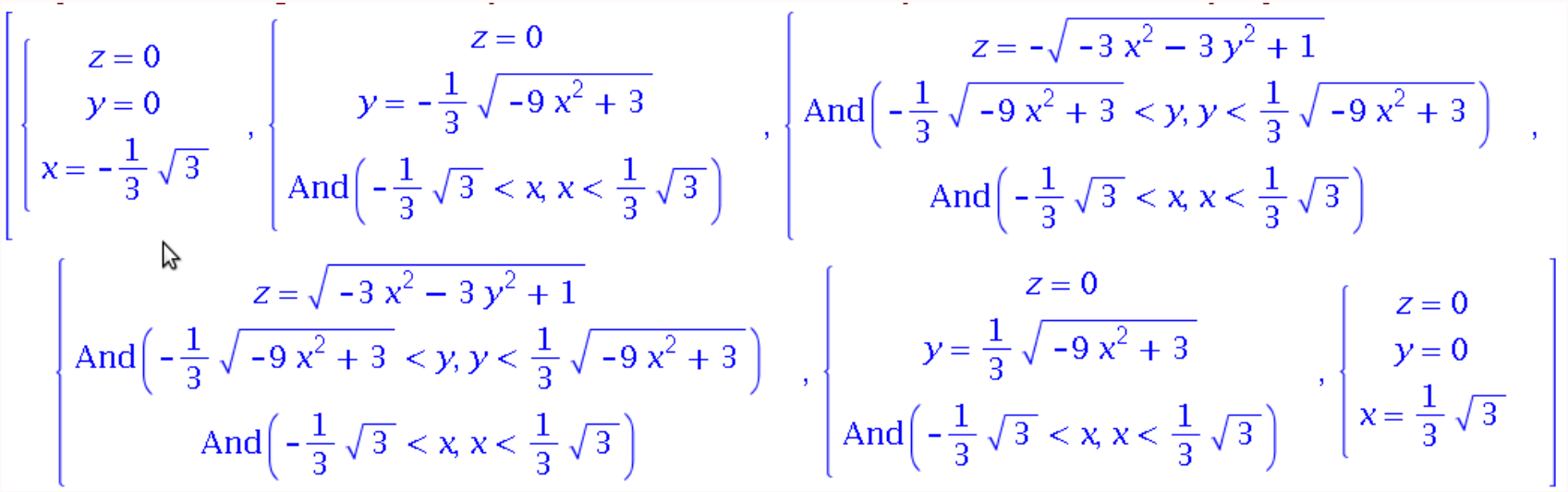
A sign-invariant CCT defined by $Dattel$ is described as below.



A sign-invariant CAD defined by $Dattel$ is described as below. The "1" in the picture is a placeholder.



The CAD cells describing the algebraic surface $Dattel = 0$.



The projection of $Dattel = 0$ on the (x, y)-space is equivalent to the following quantifier free formula.

$$(3x^2 < 1 \wedge 3y^2 + 3x^2 < 1) \vee (3x^2 - 1 = 0 \wedge y = 0) \vee (3x^2 < 1 \wedge 3y^2 + 3x^2 = 1).$$

It can be further simplified as follows.

$$3y^2 + 3x^2 \leq 1.$$

We say the CAD in this example is projection-definable since the polynomials in the tree are enough to describe the solution set.

Outline

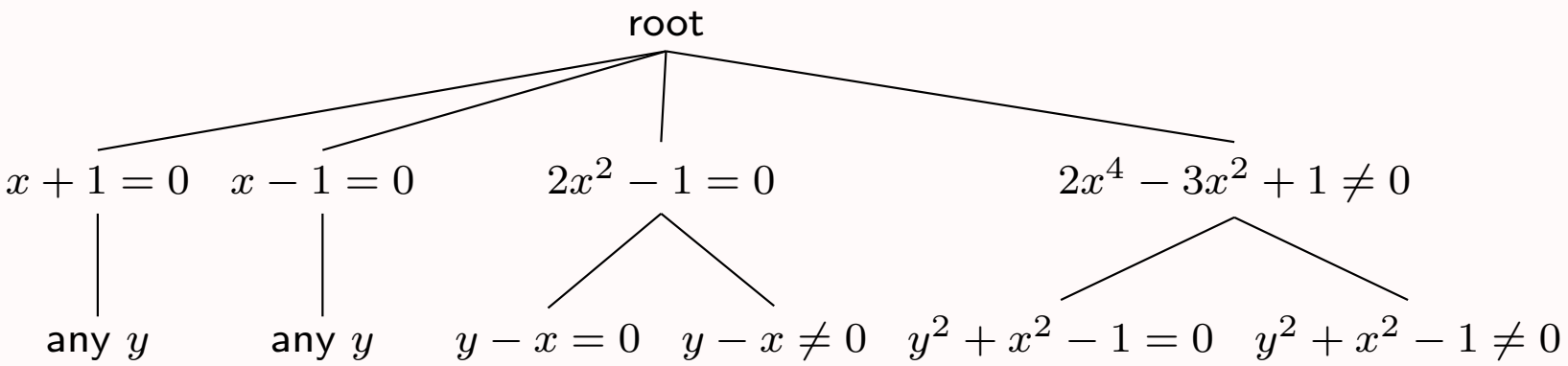
The second example

This example illustrates the case that the polynomials in the initial CCT are not enough to express the solution set.

Consider the following QE problem:

$$(\exists y)\ (x^2 + y^2 - 1 = 0) \wedge (x + y < 0) \wedge (x > -1) \wedge (x < 1).$$

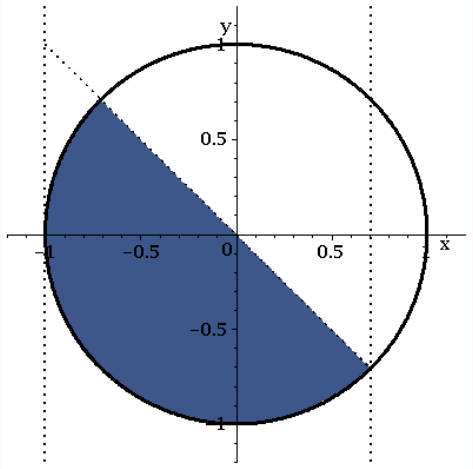
CylindricalDecompose($[x^2 + y^2 - 1 = 0, x + y \neq 0, x \neq -1, x \neq 1]$) computes the following CCT T:



- The CAD of $\mathbb{R}^1$ has the following cells, with blue ones being true cells.

$$(-\infty,-1),-1,(-1,-\frac{\sqrt{2}}{2}),-\frac{\sqrt{2}}{2},(-\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}),\frac{\sqrt{2}}{2},(\frac{\sqrt{2}}{2},1),1,(1,+\infty).$$

- The true cells describe the projection of the blue region on the x-axis, which cannot be expressed by the signs of polynomials in the CCT.



- The cells $-\frac{\sqrt{2}}{2}$ and $\frac{\sqrt{2}}{2}$ is called a conflicting pair, since they have opposite true values and all univariate polynomials in the tree have the same signs at them.
- They are derived from the path $\Gamma := [root, 2x_1^2 - 1 = 0]$ of T_1. Refine Γ w.r.t. $\mathrm{diff}(2x_1^2 - 1, x)$ generates a projection definable CAD, from where we deduce the solution
$(x_1 < 0 \wedge 0 < x_1 + 1) \vee x_1 = 0 \vee (0 < x_1 \wedge 2x_1^2 < 1)$.

Outline

Complex cylindrical tree (CCT)

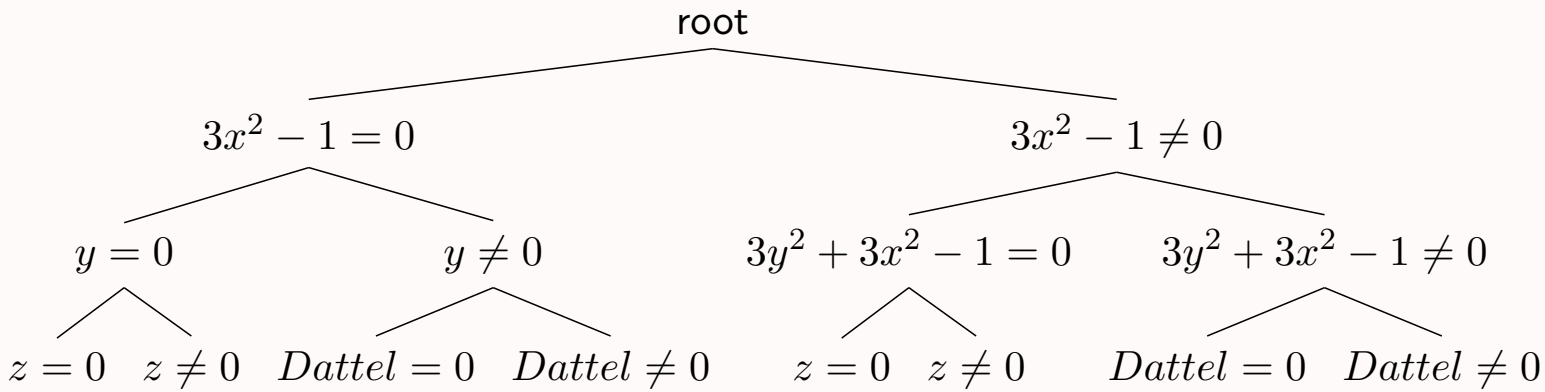
Let T be a rooted tree of height n where each node of depth i, for $i = 1, \ldots, n$, being labeled by a polynomial constraint of the type "any x_i" (with zero set defined as $\mathbb{C}^n$), or $p = 0$, or $p \neq 0$, where $p \in \mathbb{Q}[x_1, \ldots, x_i]$. We say T is a complete CCT if it satisfies:

- if $n = 1$, then either T has only one leaf labeled "any x_1", or it has $s + 1$ leaves labeled respectively $p_1 = 0, \ldots, p_s = 0, \prod_{i=1}^{s} p_i \neq 0$, where $p_1, \ldots, p_s \in \mathbb{Q}[x_1]$ are squarefree and pairwise coprime;
- if $n > 1$, then the induced subtree T_{n-1} of T is a complete CCT and for any given path Γ of T_{n-1}, either its leaf V has only one child in T of type "any x_n", or, for some $s \geq 1$, V has $s + 1$ children labeled $p_1 = 0, \ldots, p_s = 0, \prod_{i=1}^{s} p_i \neq 0$, where $p_1, \ldots, p_s \in \mathbb{Q}[\mathbf{x}]$ separate above $Z_{\mathbb{C}}(\Gamma)$ (that is for any α of $Z_{\mathbb{C}}(\Gamma)$, $\mathrm{lc}(p_i) \neq 0$ and $p_i(\alpha)$ are squarefree and coprime).

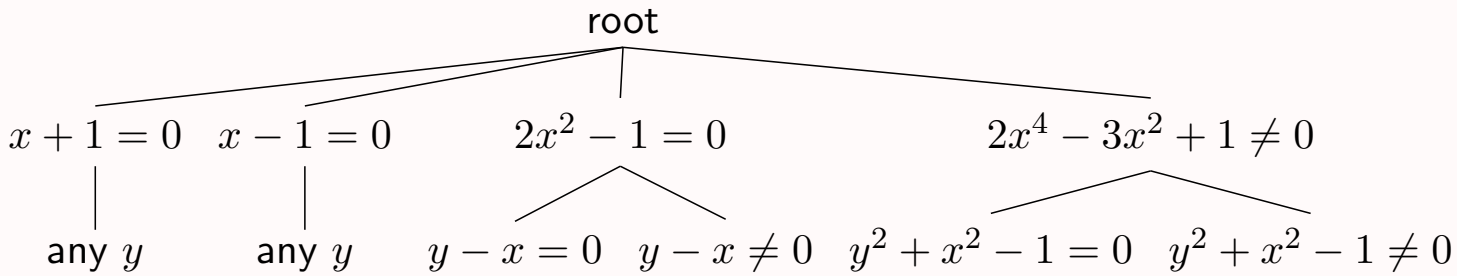
We say T is sign-invariant w.r.t. $F \subset \mathbb{Q}[\mathbf{x}]$ if for any $p \in F$ and any path Γ of T, either $Z_{\mathbb{C}}(\Gamma) \subset Z_{\mathbb{C}}(p)$ or $Z_{\mathbb{C}}(\Gamma) \cap Z_{\mathbb{C}}(p) = \emptyset$ holds.

Two previous examples

Sign-invariant CCT w.r.t. $Dattel := z^2 + 3y^2 + 3x^2 - 1$



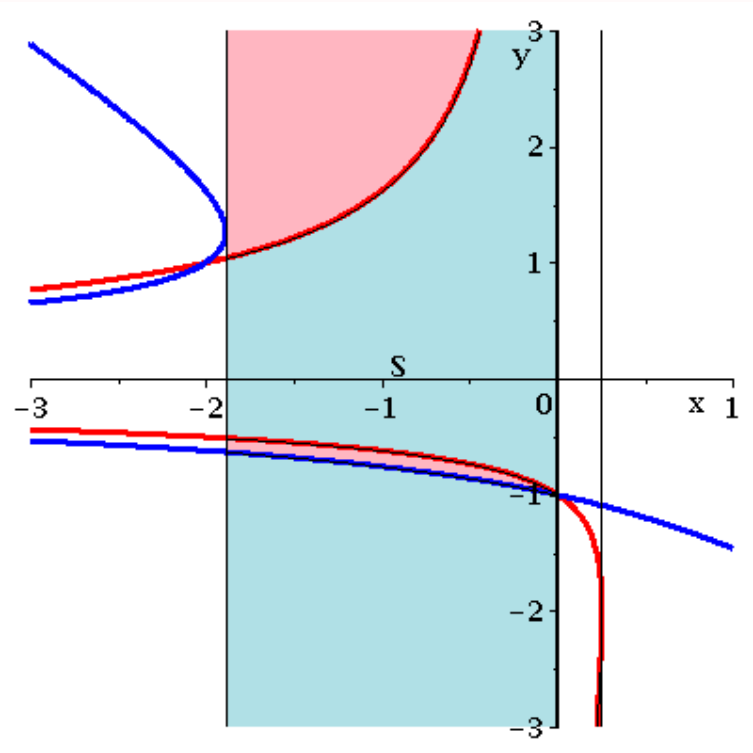
Truth-invariant CCT w.r.t. the conjunction of the constraints $\{x^2 + y^2 - 1 = 0, x + y \neq 0, x \neq -1, x \neq 1\}$



Theorem

Let $P = \{p_1, \ldots, p_r\}$ be a finite set of polynomials in $\mathbb{Q}[x_1 \prec \cdots \prec x_n]$ with the same main variable x_n. Let S be a connected semi-algebraic subset of $\mathbb{R}^{n-1}$. If P separates *above S, then each p_i is delineable on S. Moreover, the product of the $p_1, \ldots, p_r$ is also delineable on S.*

The polynomials $p := xy^2 + y + 1$, $q := y^3 + xy^2 + 1$ and their product are delineable on S.



Projection factor set

- Let T be a complete CCT in $\mathbb{Q}[\mathbf{x}]$.
- Let V be a node in T, the projection factor set of V is the set of polynomials appearing in its immediate equational siblings.
- Let Γ be a path of T. The union of the projection factor sets of all the nodes along Γ is called a *projection factor set* of Γ.
- The projection factor set of T is defined as the union of projection factor sets of all its paths.

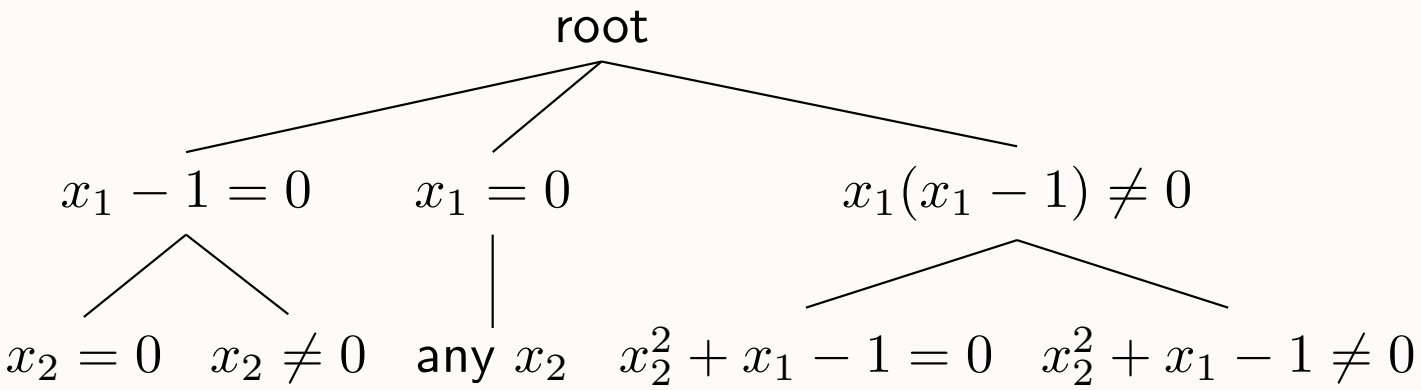
Projection definable

Let RT be a CAD tree attached with truth values. We say RT is projection definable if there exists a QFF formed by the signs of polynomials in RT's projection factor set such that the QFF defines the same zero set as the union of true CAD cells of RT.

Remark. The concept of projection factor set has different meanings for PL-CAD and RC-CAD. For the latter, the projection factor set of one path may not be sign-invariant on another path (see example in next slides).

Example

Consider the following complex cylindrical tree T.



- Let RT be a CAD tree derived from T.
- Let Γ be the right most path of T.
- The projection factor set of Γ is $\{x_1, x_1 - 1, x_2^2 + x_1 - 1\}$.
- The projection factor set of T and RT is $\{x_1, x_1 - 1, x_2^2 + x_1 - 1, x_2\}$.
- We notice that neither x_2 nor $x_2^2 + x_1 - 1$ is sign-invariant on the path of T consisting of nodes $\{root, x_1 = 0, \text{ any } x_2\}$.
- It is easy to verify that RT is projection definable.

Definition

Let F be a set of non-constant univariate polynomials in $\mathbb{R}[x]$. We say F is derivative closed (w.r.t. factorization) if for any $f \in F$, where $\deg(f) > 1$, $\mathrm{der}(f)$ is a product of some polynomials in F and some constant $c \in \mathbb{R}$.

Let $F \subset \mathbb{R}[\mathbf{x}]$ be finite. Let σ be a map from F to $\{<, >, =\}$. Let $F_\sigma := \wedge_{f \in F} f\ \sigma(f)\ 0$. Define $Z_{\mathbb{R}}(F_\sigma) := \cap_{f \in F} Z_{\mathbb{R}}(f\ \sigma(f)\ 0)$.

Lemma (Thom's Lemma Variant)

Assume that $n = 1$. If F is derivative closed (w.r.t. factorization), then the set defined by F_σ is either an empty set, a point or an open interval.

Theorem

Let C be a region of $\mathbb{R}^{n-1}$. let F be a set of polynomials in $\mathbb{Q}[x_1 \prec \cdots \prec x_n]$ with the same main variable x_n. We assume that F separates above C and that for each point α of C, the set of univariate polynomials $\{p(\alpha, x_n) \mid p \in F\}$ is derivative closed (w.r.t. factorization). Then, the set $C \times \mathbb{R}^1 \cap Z_{\mathbb{R}}(F_\sigma)$ is either empty, or a section, or a sector above C.

Conflicting pair

- Let RT_k be a CAD tree of $\mathbb{R}^k$ attached with truth values.
- Let T_k be the associated CCT of RT_k.

For $1 \leq i \leq k$, we call two distinct i-level cells C_i and D_i in the same stack an i-level conflicting pair if there exist k-level cells C and D such that

(CP_1) C_i and D_i are respectively the projections of C and D onto $\mathbb{R}^i$,

(CP_2) C and D are derived from the same path of T_k,

(CP_3) above C and D, every polynomial in their common projection factor set P has the same sign,

(CP_4) C and D have opposite attached truth values.

The algorithm for creating projection definable CAD

- Algorithm: MakeProjectionDefinable$_k(PF, RT, practical)$
  - let CPS be the set of all conflicting pairs of RT_k
  - while $CPS \neq \emptyset$ do
    1. let CP be a pair in CPS of highest level, say i
    2. let T be the associated CCT of RT
    3. let Γ be the path of T_i, where CP is derived
    4. call RefineNextChild$_i(\Gamma_{i-1}, T)$ to refine T
    5. $RT :=$ MakeSemiAlgebraic(T); AttachTruthValue(FF, RT); PropagateTruthValue(PF, RT)
    6. let CPS be the set of all conflicting pairs of RT
- Algorithm: RefineNextChild$_k(\Gamma_{k-1}, T)$
  - $V := \Gamma_{k-1}.leaf$
  - $S := \{c | c \in children(V), c$ is of the form $f = 0, \deg(f) > 1, c.derivative$ is undefined$\}$
  - if $S = \emptyset$ then return false
  - let $c \in S$ such that $\deg(f)$ is the smallest
  - $c.derivative := \mathrm{der}(f, x_k)$
  - let Γ_k be the subtree of T_k which induces Γ_{k-1}
  - While $C :=$ NextPathToDo$_k(\Gamma_k \setminus (\Gamma_{k-1} \cup c))$ do IntersectPath$_k(\mathrm{der}(f, x_k), C, T_k)$
  - return true

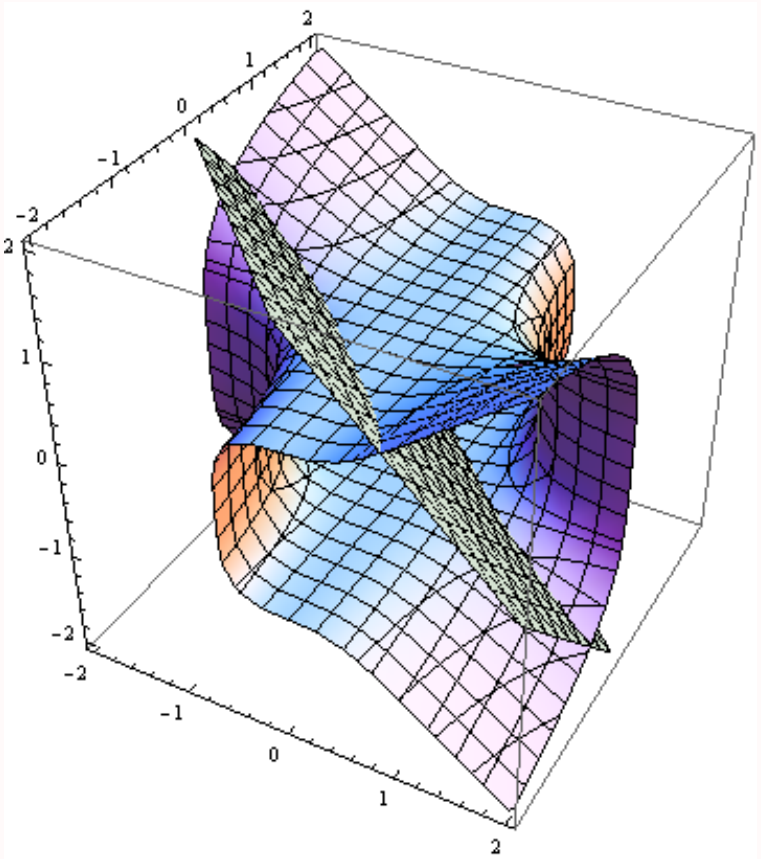
Outline

An advanced example

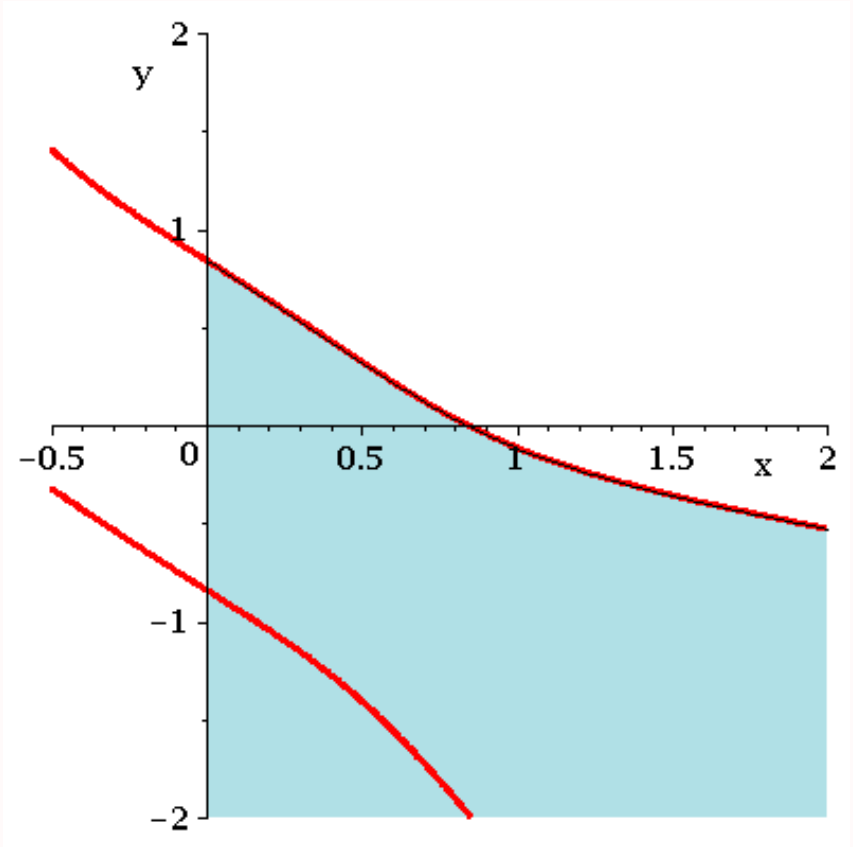
Let $f := 2z^4 + 2x^3y - 1$ and $h = x + y + z$.
Consider the following quantifier elimination problem.

$$\exists(z)(f < 0 \wedge h < 0).$$

The plots of $f = 0$ and $h = 0$ are depicted in the following figure.

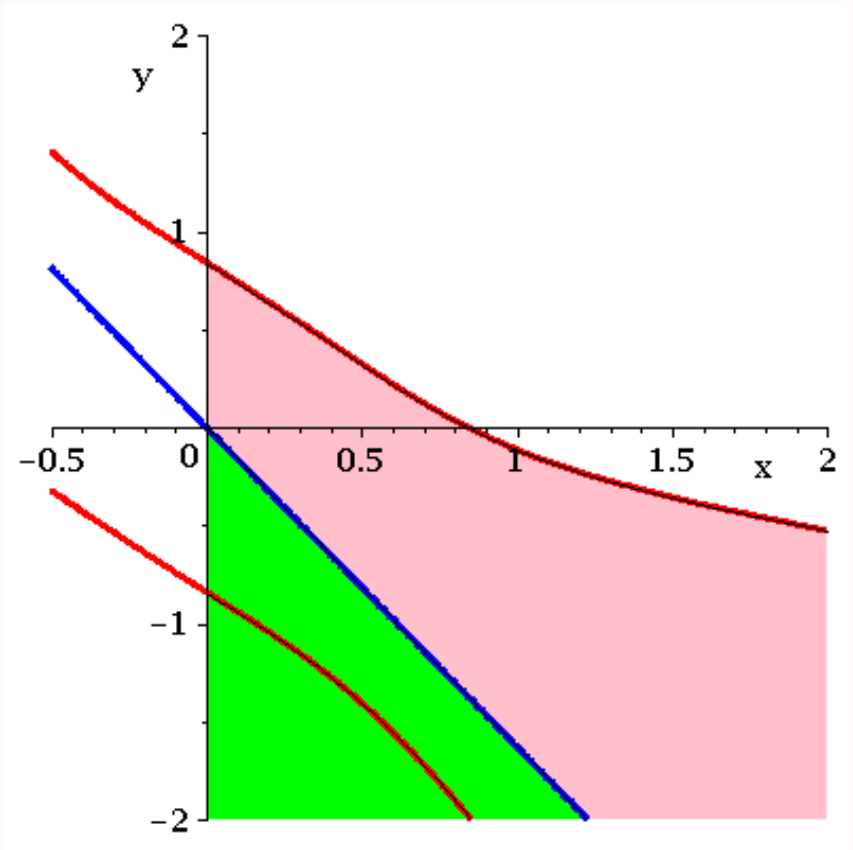


The solution set is the blue region in the following picture, where the red curve is the locus of $p := 2x^4 + 10x^3y + 12x^2y^2 + 8xy^3 + 2y^4 - 1$. The solution set is exactly the set of (x, y) such that $x > 0$ and $y < RealRoot_2(p, y)$. Apparently, this region cannot be described just by the sign of p.



To describe the blue region by a QFF, the derivative of p, namely $q := 10x^3 + 24x^2y + 24xy^2 + 8y^3$, is introduced. The locus of q is the blue curve.

Note that the blue region is the union of the green region $(x > 0 \wedge q < 0)$, the blue curve $(x > 0 \wedge q = 0)$ and the pink region $(x > 0 \wedge p < 0 \wedge q > 0)$.



Outline

The user interface of the QE procedure

We have developed the interface of our QE procedure based on the Logic package of MAPLE. The following MAPLE session shows how to use our procedure.

Example (Davenport-Heintz)

The interface:

```
> f := &E([c]), &A([b, a]), ((a=d) &and (b=c))
              &or ((a=c) &and (b=1)) &implies (a^2=b):
> QuantifierElimination(f);
              (d - 1 = 0) &or (d + 1 = 0)
```

Benchmark examples

- The efficiency of the QE procedure directly benefits that of RC-CAD.
- It was shown in ASCM 12 that RC-CAD is competitive to the state of art CAD implementations.
- We illustrate the efficiency of the QE procedure by several examples.

Neither QEPCAD nor Mathematica can solve the examples blood-coagulation-2 and MontesS10 within 1-hour time limit.

Example (blood-coagulation-2)

It takes about 6 seconds.

```
f := &E([x, y, z]), (1/200*x*s*(1 - 1/400*x)
       + y*s*(1 - 1/400*x) - 35/2*x=0)
       &and (250*x*s*(1 - 1/600*y )*(z + 3/250) - 55/2*y=0)
       &and (500*(y + 1/20*x)*(1 - 1/700*z) - 5*z=0);
QuantifierElimination(f);
true
```

Example (MontesS10)

It takes about 26 seconds.

```
f := &E([c2,s2,c1,s1]),
      (r-c1+l*(s1*s2-c1*c2)=0) &and (z-s1-l*(s1*c2+s2*c1)=0)
      &and (s1^2+c1^2-1=0) &and (s2^2+c2^2-1=0);
QuantifierElimination(f);
```

```
      2    2    2
((((-r  - z  + l  - 2 l + 1 = 0) &or

   2    2    2                      2    2    2
((l  - r  - z  - 2 l < -1) &and (-r  - z  + l  + 2 l + 1 = 0))) &or

   2    2    2                        2    2    2
((l  - r  - z  - 2 l < -1) &and (0 < -r  - z  + l  + 2 l + 1))) &or

       2    2    2                    2    2    2
((0 < -r  - z  + l  - 2 l + 1) &and (l  - r  - z  + 2 l < -1))) &or

       2    2    2                      2    2    2
((0 < -r  - z  + l  - 2 l + 1) &and (-r  - z  + l  + 2 l + 1 = 0))
```

Consider a new example on algebraic surfaces.

Example (Sattel-Dattel-Zitrus)

It takes about 3 seconds while QEPCAD cannot solve it in 30 minutes.

```
Sattel := x^2+y^2*z+z^3;
Dattel := 3*x^2+3*y^2+z^2-1;
Zitrus := x^2+z^2-y^3*(y-1)^3;
f := &E([y, z]), (Sattel=0) &and (Dattel=0) &and (Zitrus<0);
QuantifierElimination(f);
```

The output is the inequality:

$$
\begin{aligned}
&387420489\,x^{36} + 473513931\,x^{34} + 1615049199\,x^{32} \\
&-5422961745\,x^{30} + 2179233963\,x^{28} - 14860773459\,x^{26} \\
&+43317737551\,x^{24} - 45925857657\,x^{22} + 60356422059\,x^{20} \\
&-126478283472\,x^{18} + 164389796305\,x^{16} - 121571730573\,x^{14} \\
&+54842719755\,x^{12} - 16059214980\,x^{10} + 3210573925\,x^{8} \\
&-446456947\,x^{6} + 43657673\,x^{4} - 1631864\,x^{2} < 40328.
\end{aligned}
$$

Outline

Conclusion and future work

- We have proposed a complete algorithm for doing QE by RC-CAD.
- The efficiency of the QE procedure is justified by examples.
- The efficiency of the QE can benefit from that of RC-CAD and related optimizations like RC-TTICAD.
- Further work is required to get simpler output QFF.
- Further work is required to support partial cylindrical algebraic decompositions.