max planck institut
informatik

(Mostly Real) Quantifier Elimination

Thomas Sturm

AVACS Autumn School, Oldenburg, Germany, October 1, 2015

http://www.mpi-inf.mpg.de/~sturm/

Overview

Quantifier Elimination and Decision

Example (Tarski Algebra = real numbers with arithmetic and ordering)

$$\mathbb{R} \models \underbrace{\forall x \exists y (x^2 + xy + b > 0 \land x + ay^2 + b \leq 0)}_{\varphi} \longleftrightarrow \underbrace{a < 0 \land b > 0}_{\varphi'}$$

Quantifier Elimination and Decision

Example (Tarski Algebra = real numbers with arithmetic and ordering)

$$\mathbb{R} \models \underbrace{\forall x \exists y (x^2 + xy + b > 0 \land x + ay^2 + b \leq 0)}_{\varphi} \longleftrightarrow \underbrace{a < 0 \land b > 0}_{\varphi'}$$

Formally: Given 1st-order theory Θ, find **algorithm** with input φ and output φ' quantifier-free such that

$$\Theta \models \varphi \longleftrightarrow \varphi',$$

or prove that no such algorithm exists.

Important aspects: theoretical complexity, practical performance

Quantifier Elimination and Decision

Example (Tarski Algebra = real numbers with arithmetic and ordering)

$$\mathbb{R} \models \underbrace{\forall x \exists y (x^2 + xy + b > 0 \land x + ay^2 + b \leq 0)}_{\varphi} \longleftrightarrow \underbrace{a < 0 \land b > 0}_{\varphi'}$$

Formally: Given 1st-order theory Θ, find **algorithm** with input φ and output φ' quantifier-free such that

$$\Theta \models \varphi \longleftrightarrow \varphi',$$

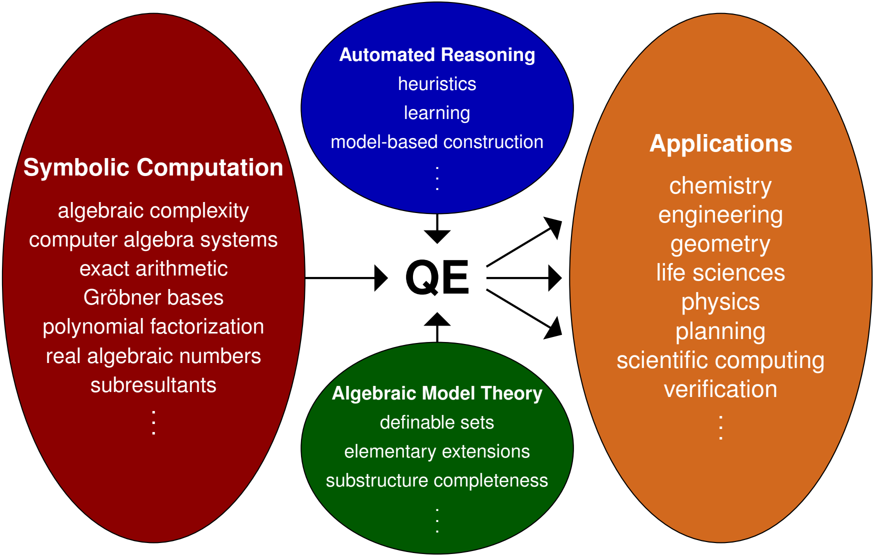
or prove that no such algorithm exists.

Important aspects: theoretical complexity, practical performance

Important Special Cases

- all variables in φ are quantified $\rightsquigarrow$ **decision problem**
- only existential quantifiers $\rightsquigarrow$ **satisfiability problem**

Quantifier Elimination-relevant Research Topics



Definitions

Syntax and Semantics

Language (= Signature): $L = (0, 1, +, -, \cdot, <, \leq, \neq, >, \geq)$

Semantics: Everything is interpreted over $\mathbb{R}$.

Syntax and Semantics

Language (= Signature): $L = (0, 1, +, -, \cdot, <, \leq, \neq, >, \geq)$

Semantics: Everything is interpreted over $\mathbb{R}$.

Important convention in algebraic model theory

There is always"$=$" which is formally not in the language.

Semantics of "$=$" is Leibniz's (second-order) definition of **equality**

$$x = y :\Longleftrightarrow \forall p\big(p(x) \longleftrightarrow p(y)\big)$$

in contrast to its first-order theory.

For convenience, define $L_{=} := L \cup \{=\}$.

max planck institut
informatik

Syntax and Semantics

Language (= Signature): $L = (0, 1, +, -, \cdot, <, \leq, \neq, >, \geq)$

Semantics: Everything is interpreted over $\mathbb{R}$.

Important convention in algebraic model theory

There is always"$=$" which is formally not in the language.

Semantics of "$=$" is Leibniz's (second-order) definition of **equality**

$$x = y :\iff \forall p\big(p(x) \longleftrightarrow p(y)\big)$$

in contrast to its first-order theory.

For convenience, define $L_= := L \cup \{=\}$.

Remark

There is no multiplicative inverse or division in L.

We do not want to deal with partial functions.

Terms and Atomic Formulas

Terms

are w.l.o.g. polynomials with integer coefficients in a recursive representation

$$t \in (\dots(((\mathbb{Z}[x_n])[x_{n-1}])\dots)[x_2])[x_1]$$

Representation is unique and isomorphic to "distributive" $\mathbb{Z}[x_1, \dots, x_n]$.

Example

$f = x_1 + (x_2 + x_3), \quad f^2 = x_1^2 + (2x_2 + 2x_3)x_1 + (x_2^2 + 2x_3x_2 + x_3^2)$

We can efficiently **reorder** such polynomials, i.e., change the main variable.

Terms and Atomic Formulas

Terms

are w.l.o.g. polynomials with integer coefficients in a recursive representation

$$t \in (\dots(((\mathbb{Z}[x_n])[x_{n-1}])\dots)[x_2])[x_1]$$

Representation is unique and isomorphic to "distributive" $\mathbb{Z}[x_1, \dots, x_n]$.

Example

$f = x_1 + (x_2 + x_3), \quad f^2 = x_1^2 + (2x_2 + 2x_3)x_1 + (x_2^2 + 2x_3x_2 + x_3^2)$

We can efficiently **reorder** such polynomials, i.e., change the main variable.

Atomic formulas (atoms) are of the form $f \; R \; 0$, where

- $R \in L_= = \{\leq, <, \neq, \geq, >, =\}$ as discussed
- f a recursive polynomial in some variables $x_1, \dots, x_n$ as above
- $L_=$ is closed under negation: For $R \in L_=$ there is $\bar{R} \in L_=$ such that

$$\mathbb{R} \models \neg(f \; R \; 0) \longleftrightarrow f \; \bar{R} \; 0.$$

Quantifier-free Formulas and First-order Formulas

First-order formulas are obtained from atomic formulas using operators

$$\text{true}, \quad \text{false}, \quad \wedge, \quad \vee, \quad \exists x, \quad \forall x, \qquad \text{where } x \text{ is a variable}$$

Further Boolean Operators

- $\longrightarrow$ and $\longleftrightarrow$ can be expressed without introducing quantifiers:

$$\alpha \longrightarrow \beta \quad \rightsquigarrow \quad \neg\alpha \vee \beta, \qquad \alpha \longleftrightarrow \beta \quad \rightsquigarrow \quad \alpha \longrightarrow \beta \wedge \beta \longrightarrow \alpha.$$

- Eliminate $\neg$ using de Morgan's law and closure property of L w.r.t. negation, e.g.:

$$\neg(x = 0 \wedge y > 0) \quad \rightsquigarrow \quad x \neq 0 \vee y \leq 0.$$

Quantifier-free Formulas and First-order Formulas

First-order formulas are obtained from atomic formulas using operators

$$\text{true}, \quad \text{false}, \quad \wedge, \quad \vee, \quad \exists x, \quad \forall x, \qquad \text{where } x \text{ is a variable}$$

Further Boolean Operators

- $\longrightarrow$ and $\longleftrightarrow$ can be expressed without introducing quantifiers:

$$\alpha \longrightarrow \beta \quad \rightsquigarrow \quad \neg\alpha \vee \beta, \qquad \alpha \longleftrightarrow \beta \quad \rightsquigarrow \quad \alpha \longrightarrow \beta \wedge \beta \longrightarrow \alpha.$$

- Eliminate $\neg$ using de Morgan's law and closure property of L w.r.t. negation, e.g.:

$$\neg(x = 0 \wedge y > 0) \quad \rightsquigarrow \quad x \neq 0 \vee y \leq 0.$$

Practical reson for restricting to $\wedge$ and $\vee$: **Simplification**

Quantifier-free formulas are first-order formulas not containing $\exists x$ or $\forall x$.

Convention: the only formulas containing true, false are true, false themselves.

Prenex Formulas

We assume w.l.o.g. that all first-order formulas are in a **prenex normal form**

$$Q_n x_n \ldots Q_1 x_1 (\psi)$$

with $Q_1, \ldots, Q_n \in \{\exists, \forall\}$ and ψ quantifier-free.

Fact

(i) For every first-order formula $\tilde{\varphi}$ there is an equivalent prenex formula

$$\varphi = Q_n x_n \ldots Q_1 (\psi).$$

(ii) φ can be efficiently computed from $\tilde{\varphi}$ such that the number of alternations in the sequence $Q_n, \ldots, Q_1$ is minimized.

Virtual Substitution

Eliminate from the Inside to the Outside

Given $\varphi = \mathrm{Q}_n x_n \dots \mathrm{Q}_1 x_1(\psi)$

- ψ is quantifier-free
- the variables of ψ are a subset of **quantified (bound) variables** $X = \{x_1, \dots, x_n\}$ and **(free) parameters** $U = \{u_1, \dots, u_m\}$, where

$$X \cap U = \emptyset.$$

We are going to eliminate $\mathrm{Q}_1 x_1$.

The rest is iteration with some optimizations to discuss later on.

We may assume that $\mathrm{Q}_1 = \exists$, because $\forall x_1 \varphi \longleftrightarrow \neg\exists x_1 \neg\varphi$.

Elimination of One Existential Quantifier

Given $\varphi = \exists x_1(\psi)$

- The variables in ψ are among x_1 and $V_1 := (X \setminus \{x_1\}) \cup U$.
- All variables from V_1 will play the same role now, say, $V_1 = \{v_1, \dots, v_k\}$.

If x_1 does not occur in ψ, then we are done.

Key Idea

- **Intuitively**, $\exists x$ is like a big disjunction over all real **numbers**.
- Could there be a finite E set of **terms** t such that

$$\mathbb{R} \models \exists x_1(\psi) \longleftrightarrow \bigvee_{t \in E} \psi[x_1/t] \quad ?$$

Modulo a couple of technical problems, there is essentially such a set.

Thought Experiment

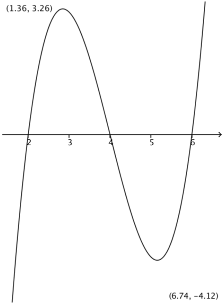
Given $\varphi = \exists x_1(\psi)$

Temporarily and only in our minds (not in any algorithm) fix

$$(v_1, \dots, v_k) := (a_1, \dots, a_l) \in \mathbb{R}^k$$

such that ψ becomes univariate in x_1.

Left hand sides of atomic formulas in ψ become univariate polynomials $f \in \mathbb{R}[x_1]$.



- Sets of satisfying values for x_1 in $f(x_1)\ R\ 0$ are

Thought Experiment

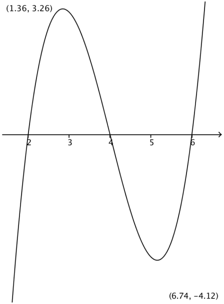
Given $\varphi = \exists x_1(\psi)$

Temporarily and only in our minds (not in any algorithm) fix

$$(v_1, \dots, v_k) := (a_1, \dots, a_l) \in \mathbb{R}^k$$

such that ψ becomes univariate in x_1.

Left hand sides of atomic formulas in ψ become univariate polynomials $f \in \mathbb{R}[x_1]$.



- Sets of satisfying values for x_1 in $f(x_1)\ R\ 0$ are **finite unions of intervals** $[b_1, b_2]$, (b_1, b_2), $(b_1, b_2]$, $[b_1, b_2)$, where b_1, $b_2 \in \mathbb{R} \cup \{\infty\}$.

max planck institut
informatik

Thought Experiment

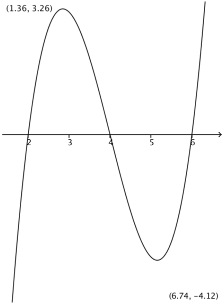
Given $\varphi = \exists x_1(\psi)$

Temporarily and only in our minds (not in any algorithm) fix

$$(v_1, \dots, v_k) := (a_1, \dots, a_l) \in \mathbb{R}^k$$

such that ψ becomes univariate in x_1.

Left hand sides of atomic formulas in ψ become univariate polynomials $f \in \mathbb{R}[x_1]$.



- Sets of satisfying values for x_1 in $f(x_1)\ R\ 0$ are **finite unions of intervals** $[b_1, b_2]$, (b_1, b_2), $(b_1, b_2]$, $[b_1, b_2)$, where $b_1, b_2 \in \mathbb{R} \cup \{\infty\}$.
- if $b_i \in \mathbb{R}$, then $f(b_i) = 0$
- Set of satisfying values for x_1 in ψ has the same form.

Thought Experiment

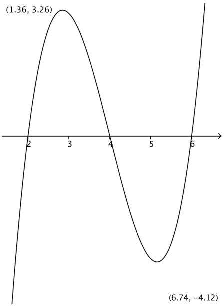
Given $\varphi = \exists x_1(\psi)$

Temporarily and only in our minds (not in any algorithm) fix

$$(v_1, \dots, v_k) := (a_1, \dots, a_l) \in \mathbb{R}^k$$

such that ψ becomes univariate in x_1.

Left hand sides of atomic formulas in ψ become univariate polynomials $f \in \mathbb{R}[x_1]$.



- Sets of satisfying values for x_1 in $f(x_1)\ R\ 0$ are **finite unions of intervals** $[b_1, b_2]$, (b_1, b_2), $(b_1, b_2]$, $[b_1, b_2)$, where b_1, $b_2 \in \mathbb{R} \cup \{\infty\}$.
- if $b_i \in \mathbb{R}$, then $f(b_i) = 0$
- Set of satisfying values for x_1 in ψ has the same form. $\wedge$ is cut and $\vee$ is intersection of satisfying sets.

Thought Experiment

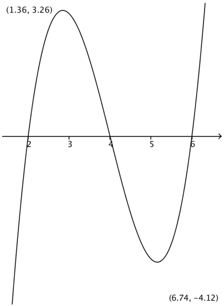
Given $\varphi = \exists x_1(\psi)$

Temporarily and only in our minds (not in any algorithm) fix

$$(v_1, \ldots, v_k) := (a_1, \ldots, a_l) \in \mathbb{R}^k$$

such that ψ becomes univariate in x_1.

Left hand sides of atomic formulas in ψ become univariate polynomials $f \in \mathbb{R}[x_1]$.



- Sets of satisfying values for x_1 in $f(x_1)\ R\ 0$ are **finite unions of intervals** $[b_1, b_2]$, (b_1, b_2), $(b_1, b_2]$, $[b_1, b_2)$, where b_1, $b_2 \in \mathbb{R} \cup \{\infty\}$.
- if $b_i \in \mathbb{R}$, then $f(b_i) = 0$
- Set of satisfying values for x_1 in ψ has the same form. $\wedge$ is cut and $\vee$ is intersection of satisfying sets.
- Idea: E = all b_2 or $b_2 - \varepsilon$ and ∞.

Elimination Sets

Given $\varphi = \exists x_1(\psi)$

Supersets of the zeros of the left hand side terms

$$f \in (\dots(((\mathbb{Z}[v_1])[v_2])\dots)[v_k])[x_1]$$

can be computed **symbolically** and **uniformly**.

Example

$f = a(v_1 \dots, v_k)x_1^2 + b(v_1, \dots, v_k)x_1 + c(v_1, \dots, v_k)$ yields candidate solutions

$$\underbrace{(-b \pm \sqrt{b^2 - 4ac})/2a}_{t} \text{ for } \underbrace{a \neq 0 \wedge b^2 - 4ac \geq 0}_{\gamma}, \qquad \underbrace{-c/b}_{t} \text{ for } \underbrace{a = 0 \wedge b \neq 0}_{\gamma}.$$

Elimination Sets

Given $\varphi = \exists x_1(\psi)$

Supersets of the zeros of the left hand side terms

$$f \in (\dots(((\mathbb{Z}[v_1])[v_2])\dots)[v_k])[x_1]$$

can be computed **symbolically** and **uniformly**.

Example

$f = a(v_1 \dots, v_k)x_1^2 + b(v_1, \dots, v_k)x_1 + c(v_1, \dots, v_k)$ yields candidate solutions

$$\underbrace{\left(-b \pm \sqrt{b^2 - 4ac}\right)/2a}_{t} \text{ for } \underbrace{a \neq 0 \land b^2 - 4ac \geq 0}_{\gamma}, \qquad \underbrace{-c/b}_{t} \text{ for } \underbrace{a = 0 \land b \neq 0}_{\gamma}.$$

An **elimination set** E for x_1 and ψ is a finite set of pairs (γ, t) such that

$$\mathbb{R} \models \exists x_1(\psi) \longleftrightarrow \bigvee_{(\gamma,t)\in E} \gamma \land \psi[x_1 /\!/ t].$$

Virtual Substitution

Given $\varphi = \exists x_1(\psi)$ and E such that $\mathbb{R} \models \exists x_1(\psi) \longleftrightarrow \bigvee_{(\gamma,t)\in E} \gamma \wedge \psi[x_1 /\!/ t]$.

Remaining Problem

t contain $/$, $\sqrt{\cdot}$, ∞, ε, $\dots$, which are not in our language L.

Solution: Virtual Substitution

$$[x /\!/ t] : \text{atomic formulas} \rightarrow \text{quantifier-free formulas}$$

Virtual Substitution

Given $\varphi = \exists x_1(\psi)$ and E such that $\mathbb{R} \models \exists x_1(\psi) \longleftrightarrow \bigvee_{(\gamma,t)\in E} \gamma \wedge \psi[x_1 /\!/ t]$.

Remaining Problem

t contain $/$, $\sqrt{\cdot}$, ∞, ε, $\ldots$, which are not in our language L.

Solution: Virtual Substitution

$$[x /\!/ t] : \text{atomic formulas} \rightarrow \text{quantifier-free formulas}$$

And beyond degree 2?

- Method generalizes to arbitrary degrees (in principle long known).
- first implementation will be available this year (PhD thesis by M. Košta).
- For higher degrees, t will be way more abstract.

Virtual Substitution

Given $\varphi = \exists x_1(\psi)$ and E such that $\mathbb{R} \models \exists x_1(\psi) \longleftrightarrow \bigvee_{(\gamma,t)\in E} \gamma \wedge \psi[x_1 /\!/ t]$.

Remaining Problem

t contain $/$, $\sqrt{\cdot}$, ∞, ε, $\ldots$, which are not in our language L.

Solution: Virtual Substitution

$$[x /\!/ t] : \text{atomic formulas} \longrightarrow \text{quantifier-free formulas}$$

And beyond degree 2?

- Method generalizes to arbitrary degrees (in principle long known).
- first implementation will be available this year (PhD thesis by M. Košta).
- For higher degrees, t will be way more abstract.

Important

In practice, good simplification of quantifier-free (intermediate) results is crucial!

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x]$, f_i, g_i, $g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$(f_1 x + f_0 \leq 0)\left[x /\!/ \tfrac{g_1}{g_2}\right] \equiv f_1 \frac{g_1}{g_2} + f_0 \leq 0 \equiv f_1 g_1 g_2 + f_0 g_2^2 \leq 0$$

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x], \quad f_i, \quad g_i, \quad g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$(f_1 x + f_0 \leq 0)\left[x /\!/ \tfrac{g_1}{g_2}\right] \equiv f_1 \tfrac{g_1}{g_2} + f_0 \leq 0 \equiv f_1 g_1 g_2 + f_0 g_2^2 \leq 0$$

Formal solutions of quadratic equations

$$\left(f = 0\right)\left[x /\!/ \tfrac{g_1 + g_2\sqrt{g_3}}{g_4}\right] \equiv \frac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0$$

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x], \quad f_i, \quad g_i, \quad g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$\left(f_1 x + f_0 \le 0\right)\left[x /\!/ \tfrac{g_1}{g_2}\right] \;\equiv\; f_1 \tfrac{g_1}{g_2} + f_0 \le 0 \;\equiv\; f_1 g_1 g_2 + f_0 g_2^2 \le 0$$

Formal solutions of quadratic equations

$$\Big(f = 0\Big)\Big[x /\!/ \tfrac{g_1 + g_2\sqrt{g_3}}{g_4}\Big] \;\equiv\; \frac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0 \;\equiv\; {g_1^*}^2 - {g_2^*}^2 g_3 = 0 \wedge g_1^* g_2^* \le 0$$

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x], \quad f_i, \quad g_i, \quad g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$(f_1 x + f_0 \leq 0)\left[x /\!\!/ \tfrac{g_1}{g_2}\right] \equiv f_1 \tfrac{g_1}{g_2} + f_0 \leq 0 \equiv f_1 g_1 g_2 + f_0 g_2^2 \leq 0$$

Formal solutions of quadratic equations

$$\left(f = 0\right)\left[x /\!\!/ \tfrac{g_1 + g_2\sqrt{g_3}}{g_4}\right] \equiv \frac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0 \equiv {g_1^*}^2 - {g_2^*}^2 g_3 = 0 \wedge g_1^* g_2^* \leq 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} \leq 0 \equiv ({g_1^*}^2 - {g_2^*}^2 g_3 \geq 0 \wedge g_1^* g_4^* \leq 0) \vee ({g_1^*}^2 - {g_2^*}^2 g_3 \leq 0 \wedge g_2^* g_4^* \leq 0)$$

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x]$, f_i, g_i, $g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$(f_1 x + f_0 \leq 0)\left[x /\!/ \tfrac{g_1}{g_2}\right] \equiv f_1 \tfrac{g_1}{g_2} + f_0 \leq 0 \equiv f_1 g_1 g_2 + f_0 g_2^2 \leq 0$$

Formal solutions of quadratic equations

$$\left(f = 0\right)\left[x /\!/ \tfrac{g_1 + g_2\sqrt{g_3}}{g_4}\right] \equiv \frac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0 \equiv {g_1^*}^2 - {g_2^*}^2 g_3 = 0 \wedge g_1^* g_2^* \leq 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} \leq 0 \equiv ({g_1^*}^2 - {g_2^*}^2 g_3 \geq 0 \wedge g_1^* g_4^* \leq 0) \vee ({g_1^*}^2 - {g_2^*}^2 g_3 \leq 0 \wedge g_2^* g_4^* \leq 0)$$

Infinity

$$(f_2 x^2 + f_1 x + f_0 < 0)[x /\!/ \infty] \equiv f_2 < 0 \vee (f_2 = 0 \wedge f_1 < 0) \vee (f_2 = 0 \wedge f_1 = 0 \wedge f_0 < 0)$$

Virtual Substitution by Example

Conventions: $f \in \mathbb{Z}[\mathbf{y}][x]$, f_i, g_i, $g_i^* \in \mathbb{Z}[\mathbf{y}]$

Quotients

$$(f_1 x + f_0 \le 0)\,[x /\!/ \tfrac{g_1}{g_2}] \;\equiv\; f_1 \tfrac{g_1}{g_2} + f_0 \le 0 \;\equiv\; f_1 g_1 g_2 + f_0 g_2^2 \le 0$$

Formal solutions of quadratic equations

$$\Big(f = 0\Big)\Big[x /\!/ \tfrac{g_1 + g_2\sqrt{g_3}}{g_4}\Big] \;\equiv\; \frac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} = 0 \;\equiv\; {g_1^*}^2 - {g_2^*}^2 g_3 = 0 \wedge g_1^* g_2^* \le 0$$

$$\tfrac{g_1^* + g_2^*\sqrt{g_3}}{g_4^*} \le 0 \;\equiv\; ({g_1^*}^2 - {g_2^*}^2 g_3 \ge 0 \wedge g_1^* g_4^* \le 0) \vee ({g_1^*}^2 - {g_2^*}^2 g_3 \le 0 \wedge g_2^* g_4^* \le 0)$$

Infinity

$$(f_2 x^2 + f_1 x + f_0 < 0)[x /\!/ \infty] \equiv f_2 < 0 \vee (f_2 = 0 \wedge f_1 < 0) \vee (f_2 = 0 \wedge f_1 = 0 \wedge f_0 < 0)$$

Positive infinitesimals

$$(3x^2 + 6x - 3 > 0)[x /\!/ t - \varepsilon] \equiv 3t^2 + 6t - 3 > 0 \vee (3t^2 + 6t - 3 = 0 \wedge 6t + 6 \le 0)$$

Elimination of Several Existential Quantifiers by Block

Back to the bigger picture

$$\ldots\forall^*\exists^*\forall^*\exists^*\exists x_1(\psi) \quad \rightsquigarrow \quad \ldots\forall^*\exists^*\forall^*\exists^* \bigvee_{(\gamma,t)\in E} \gamma \wedge \psi[x_1 /\!/ t]$$

Disjunction $\bigvee$ is compatible with existential quantifiers $\exists^*$:

$$\ldots\forall^*\exists^*\forall^*\exists^* \bigvee_{(\gamma,t)\in E} \gamma \wedge \psi[x_1 /\!/ t] \quad \rightsquigarrow \quad \ldots\forall^*\exists^*\forall^* \bigvee_{(\gamma,t)\in E} \exists^*(\gamma \wedge \psi[x_1 /\!/ t])$$

Elimination of Several Existential Quantifiers by Block

Back to the bigger picture

$$\ldots\forall^*\exists^*\forall^*\exists^*\exists x_1(\psi) \quad\rightsquigarrow\quad \ldots\forall^*\exists^*\forall^*\exists^* \bigvee_{(\gamma,t)\in E} \gamma \land \psi[x_1 /\!\!/ t]$$

Disjunction $\bigvee$ is compatible with existential quantifiers $\exists^*$:

$$\ldots\forall^*\exists^*\forall^*\exists^* \bigvee_{(\gamma,t)\in E} \gamma \land \psi[x_1 /\!\!/ t] \quad\rightsquigarrow\quad \ldots\forall^*\exists^*\forall^* \bigvee_{(\gamma,t)\in E} \exists^*(\gamma \land \psi[x_1 /\!\!/ t])$$

Effect

- more local substitution of test points With the elimination of the next quantifiers
- even improves upper bound on asymptotic worst-case complexity

Complexity of Virtual Substitution

Upper bound on asymptotic worst-case complexity

doubly exponential in the input word length (and thus optimal)

More precisely

doubly exponential	in # quantifier alternations
singly exponential	in # quantifiers **thanks to elimination by block**
polynomial	in # parameters (= unquantified variables)
polynomial	in # atomic formulas

particularly good for

low degrees and many parameters

For comparision: Cylindrical Algberaic Decomposition (CAD)

[Collins 1973, Hong, Brown, . . .] doubly exponential in the number of **all** variables

For comparison: Asymptotically fast procedures

[Renegar, Basu–Pollack–Roy, Grigoriev, . . .] no practical relevance (so far)

Variants of Quantifier Elimination

Extended Quantifier Elimination

Generalize $\exists x \varphi \longleftrightarrow \bigvee_{(\gamma,t)\in E} \gamma \wedge \varphi[t/\!/x]$ to $\exists x\varphi \rightsquigarrow \begin{bmatrix} \vdots & \vdots \\ \gamma \wedge \varphi[t/\!/x] & x = t \\ \vdots & \vdots \end{bmatrix}$

Simple example revisited

$$\varphi \equiv \exists x(ax^2 + bx + c = 0) \rightsquigarrow \begin{bmatrix} a \neq 0 \wedge b^2 - 4ac \geq 0 & x = \dfrac{-b - \sqrt{b^2 - 4ac}}{2a} \\ a = 0 \wedge b \neq 0 & x = -\dfrac{c}{b} \\ a = 0 \wedge b = 0 \wedge c = 0 & x = \infty_1 \end{bmatrix}$$

Semantics (for fixed parameters)

Whenever some left hand side condition holds, then $\exists x\varphi$ holds and the corresponding right hand side term is **one** sample solution.

[M. Kosta, T.S., A. Dolzmann, J. Symb. Comput. 2016]

For fixed choices of parameters, standard values can be efficiently computed for all ∞_i and ε_i in a post-processing step.

Generic Quantifier Elimination

Collect negated equations from the γ in a global theory Θ:

$$E = \{\dots, (s \neq 0 \wedge \gamma', t), \dots\} \quad \rightsquigarrow \quad \Theta = \{\dots, s \neq 0, \dots\},\ E = \{\dots, (\gamma', t), \dots\}$$

Semantics

φ' is correct for all choices of parameters satisfying Θ:

$$\bigwedge \Theta \longrightarrow (\varphi' \longleftrightarrow \varphi).$$

Important observation

exception set has a lower dimension than the parameter space

Simple example revisited

$$\varphi \equiv \exists x(ax^2 + bx + c = 0) \quad \rightsquigarrow \quad \Theta = \{a \neq 0\}, \quad \varphi' \equiv b^2 - 4ac \geq 0$$

Software

Redlog and Reduce

Everything discussed here is available in our computer logic system Redlog:

`http://www.redlog.eu`

- interactive system, QE and decision for many domains, normal forms, simplification, construction and decomposition of large formulas, ...
- interfaces to Qepcad B, Gurobi, Mathematica, Z3, ...
- more than 300 citations of applications in the literature:

  geometry, verification, chemistry, life sciences, physics and engineering, scientific computation, geometry and planning, ...
- Redlog development since 1992 as part of the CAS Reduce [Hearn, 1968]
- Reduce/Redlog open-source (free-BSD) on Sourceforge since 12/2008

  `http://reduce-algebra.sourceforge.net`
- 48,318 downloads since 12/2008 (7,496 in 2014), 500+ SVN commits per year

Further Theories in Redlog

Integers (AAECC 2007, CASC 2007, CASC 2009)

- Presburger Arithmetic
- weak quantifier elimination for the full linear theory
- weak quantifier elimination also for higher degrees (special cases)

Mixed Real-Integer (Weispfenning at ISSAC 1999)

- experimental

Complex Numbers (using Comprehensive Gröbner Bases)

- language of rings only

Differential Algebras (CASC 2004)

- language of rings with unary differential operator
- computation in differentially closed field (A. Robinson, Blum)

Further Theories in Redlog

Padic Numbers (JSC 2000, ISSAC 1999, CASC 2001)

- linear formulas over p-adic fields for p prime
- optionally uniform in p
- used e.g. for solving parametric systems of congruences over the integers

Terms (CASC 2002)

- Malcev-type term algebras (with functions instead of relations)

Queues (C. Straßer at RWCA 2006)

- two-sided queues over the other theories (2-sorted)
- Implemented at present for queues of reals

Propositional Formulas (CASC 2003, ISSAC 2010)

- generalization of SAT solving
- quantified propositional calculus, i.e., parametric QSAT (aka QBF) solving

Some Other Software

- Qepcad B (Hong and Brown)
  is the reference implementation for cylindrical algebraic decomposition (CAD).
- The computer algebra system Mathematica
  has real QE: essentially CAD + virtual substitution for preprocessing.
- The computer algebra system Maple
  has been used in recent research on CAD (Davenport et al.)
- The computer algebra system Risa/Asir (originally by Fujitsu)
  has QE by virtual substitution (TS, 1996)
- Some prototypes in Japan
  based on comprehensive Gröbner bases (Sato et al.)
  or Sturm–Habicht sequences (Anai et al. in Matlab)
- Specialized implementations of CAD in SMT solvers (z3)
- Specialized implementations of virtual substitutions for SMT (SMT-RAT)

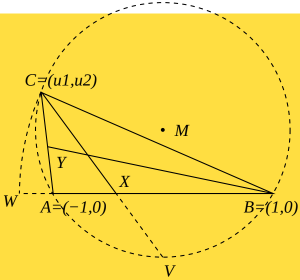
Applications in Geometry and Verification

Variant of the Steiner–Lehmus-Theorem

[J. Autom. Reasoning 1998 – Joint work with A. Dolzmann, V. Weispfenning]

The longer bisector goes to the shorter side

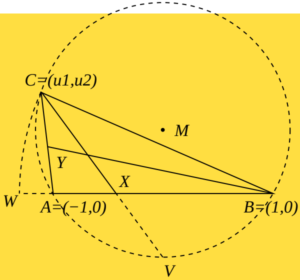
$$
\begin{aligned}
h_1 &\equiv u_2 \geq 0 \wedge x_1 \geq 0 \\
h_2 &\equiv r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2 \\
h_3 &\equiv x_2 \leq 0 \wedge r^2 = (x_2 - x_1)^2 \\
h_4 &\equiv u_1 x_2 + u_2 x_3 - x_2 x_3 = 0 \\
h_5 &\equiv x_4 \leq 1 \wedge (x_4 - 1)^2 = (u_1 - 1)^2 + u_2^2 \\
h_7 &\equiv (-1 - u_1)^2 + u_2^2 < 2^2 \\
h_6 &\equiv (x_4 - x_5)^2 + x_6^2 = (u_1 - x_5)^2 + (u_2 - x_6)^2 \wedge u_1 x_6 - u_2 x_5 - u_2 + x_6 = 0 \\
g &\equiv (u_1 - x_3)^2 + u_2^2 < (x_5 - 1)^2 + x_6^2
\end{aligned}
$$



Variant of the Steiner–Lehmus-Theorem

[J. Autom. Reasoning 1998 – Joint work with A. Dolzmann, V. Weispfenning]

The longer bisector goes to the shorter side



$$
\begin{aligned}
h_1 &\equiv u_2 \geq 0 \wedge x_1 \geq 0 \\
h_2 &\equiv r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2 \\
h_3 &\equiv x_2 \leq 0 \wedge r^2 = (x_2 - x_1)^2 \\
h_4 &\equiv u_1 x_2 + u_2 x_3 - x_2 x_3 = 0 \\
h_5 &\equiv x_4 \leq 1 \wedge (x_4 - 1)^2 = (u_1 - 1)^2 + u_2^2 \\
h_7 &\equiv (-1 - u_1)^2 + u_2^2 < 2^2 \\
h_6 &\equiv (x_4 - x_5)^2 + x_6^2 = (u_1 - x_5)^2 + (u_2 - x_6)^2 \wedge u_1 x_6 - u_2 x_5 - u_2 + x_6 = 0 \\
g &\equiv (u_1 - x_3)^2 + u_2^2 < (x_5 - 1)^2 + x_6^2
\end{aligned}
$$

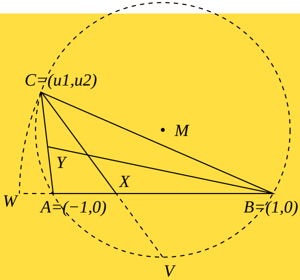
- $\varphi \equiv \forall x_6 \forall x_5 \forall x_4 \forall x_3 \forall x_2 \forall x_1 \forall r \left(\bigwedge_{i=1}^{7} h_i \longrightarrow g \right)$

Variant of the Steiner–Lehmus-Theorem

[J. Autom. Reasoning 1998 – Joint work with A. Dolzmann, V. Weispfenning]

The longer bisector goes to the shorter side

$$
\begin{aligned}
h_1 &\equiv u_2 \geq 0 \wedge x_1 \geq 0\\
h_2 &\equiv r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2\\
h_3 &\equiv x_2 \leq 0 \wedge r^2 = (x_2 - x_1)^2\\
h_4 &\equiv u_1 x_2 + u_2 x_3 - x_2 x_3 = 0\\
h_5 &\equiv x_4 \leq 1 \wedge (x_4 - 1)^2 = (u_1 - 1)^2 + u_2^2\\
h_7 &\equiv (-1 - u_1)^2 + u_2^2 < 2^2\\
h_6 &\equiv (x_4 - x_5)^2 + x_6^2 = (u_1 - x_5)^2 + (u_2 - x_6)^2 \wedge u_1 x_6 - u_2 x_5 - u_2 + x_6 = 0\\
g &\equiv (u_1 - x_3)^2 + u_2^2 < (x_5 - 1)^2 + x_6^2
\end{aligned}
$$



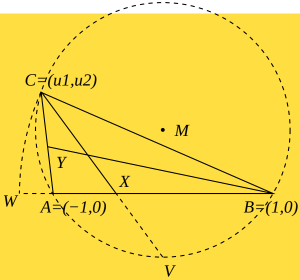
- $\varphi \equiv \forall x_6 \forall x_5 \forall x_4 \forall x_3 \forall x_2 \forall x_1 \forall r \left(\bigwedge_{i=1}^{7} h_i \longrightarrow g \right)$
- Generic QE (1.1 s): φ' 231 atomic formulas, $\Theta = \{\underbrace{u_1^2 - 2u_1 + u_2^2 - 3 \neq 0}_{(u_1-1)^2+u_2^2 \neq 4}, u_1 \neq 0, u_2 \neq 0\}$.

Variant of the Steiner–Lehmus-Theorem

[J. Autom. Reasoning 1998 – Joint work with A. Dolzmann, V. Weispfenning]

The longer bisector goes to the shorter side

$$
\begin{aligned}
h_1 &\equiv u_2 \geq 0 \land x_1 \geq 0 \\
h_2 &\equiv r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2 \\
h_3 &\equiv x_2 \leq 0 \land r^2 = (x_2 - x_1)^2 \\
h_4 &\equiv u_1 x_2 + u_2 x_3 - x_2 x_3 = 0 \\
h_5 &\equiv x_4 \leq 1 \land (x_4 - 1)^2 = (u_1 - 1)^2 + u_2^2 \\
h_7 &\equiv (-1 - u_1)^2 + u_2^2 < 2^2 \\
h_6 &\equiv (x_4 - x_5)^2 + x_6^2 = (u_1 - x_5)^2 + (u_2 - x_6)^2 \land u_1 x_6 - u_2 x_5 - u_2 + x_6 = 0 \\
g &\equiv (u_1 - x_3)^2 + u_2^2 < (x_5 - 1)^2 + x_6^2
\end{aligned}
$$



- $\varphi \equiv \forall x_6 \forall x_5 \forall x_4 \forall x_3 \forall x_2 \forall x_1 \forall r \left(\bigwedge_{i=1}^{7} h_i \longrightarrow g \right)$
- Generic QE (1.1 s): φ' 231 atomic formulas, $\Theta = \{\underbrace{u_1^2 - 2u_1 + u_2^2 - 3 \neq 0}_{(u_1-1)^2+u_2^2\neq 4}, u_1 \neq 0, u_2 \neq 0\}$.
- CAD (0.9 s): $\forall u_1 \forall u_2 (\bigwedge \Theta \longrightarrow \varphi')$ ✓

Collision Avoidance with Adaptive Cruise Control

[ISSAC 2011 – Joint Work with A. Tiwari @SRI]

System dynamics

$$\begin{aligned}
\dot{v}_f &= a_f \in [-5, 2] && \textit{velocity and accelleration of leading car} \\
\dot{v} &= a \in [-5, 2] && \textit{velocity and accelleration of rear car} \\
\dot{\mathrm{gap}} &= v_f - v && \\
\dot{a} &= -3a - 3(v - v_f) + (\mathrm{gap} - (v + 10)) && \textit{control law for rear car}
\end{aligned}$$

Initial states and safe states

$$\begin{aligned}
\mathrm{Init} &\equiv \mathrm{gap} = 10 \wedge a = 0 \wedge v_f = c_1 \wedge v = c_2 \\
\mathrm{Safe} &\equiv \mathrm{gap} > 0
\end{aligned}$$

Collision Avoidance with Adaptive Cruise Control

[ISSAC 2011 – Joint Work with A. Tiwari @SRI]

System dynamics

$$\begin{aligned} \dot{v}_f &= a_f \in [-5, 2] && \textit{velocity and accelleration of leading car} \\ \dot{v} &= a \in [-5, 2] && \textit{velocity and accelleration of rear car} \\ \dot{\text{gap}} &= v_f - v && \\ \dot{a} &= -3a - 3(v - v_f) + (\text{gap} - (v + 10)) && \textit{control law for rear car} \end{aligned}$$

Initial states and safe states

$$\begin{aligned} \text{Init} &\equiv \text{gap} = 10 \wedge a = 0 \wedge v_f = c_1 \wedge v = c_2 \\ \text{Safe} &\equiv \text{gap} > 0 \end{aligned}$$

Certificate-based approach to find a set Inv such that

1. Init $\subseteq$ Inv
2. Inv $\subseteq$ Safe
3. System dynamics cannot cause the system to leave Inv.

Collision Avoidance with Adaptive Cruise Control

Linear ansatz

$$\begin{aligned} \mathrm{Inv} &\equiv p \geq 0 \quad \text{where} \quad p := c_3 v + c_4 v_f + c_5 a + \mathrm{gap} + c_6 \\ \mathrm{Inv}' &\equiv -5 \leq a \leq 2 \ \wedge\ -5 \leq a_f \leq 2 \ \wedge\ v \geq 0 \ \wedge\ v_f \geq 0 \end{aligned}$$

Certificate as a formula

$\exists c_3 \exists c_4 \exists c_5 \exists c_6 \forall v \forall v_f \forall \mathrm{gap} \forall a \forall a_f (\varphi_1 \wedge \varphi_2 \wedge \varphi_3)$

$$\begin{aligned} \text{where} \quad \varphi_1 &\equiv \mathrm{Init} \wedge \mathrm{Inv}' \longrightarrow \mathrm{Inv} \\ \varphi_2 &\equiv \mathrm{Inv} \wedge \mathrm{Inv}' \longrightarrow \mathrm{Safe} \\ \varphi_3 &\equiv p = 0 \wedge \mathrm{Inv}' \longrightarrow \dot{p} \geq 0 \end{aligned}$$

Collision Avoidance with Adaptive Cruise Control

Linear ansatz

$$\begin{aligned} \mathrm{Inv} &\equiv p \geq 0 \quad \text{where} \quad p := c_3 v + c_4 v_f + c_5 a + \mathrm{gap} + c_6 \\ \mathrm{Inv}' &\equiv -5 \leq a \leq 2 \ \wedge\ -5 \leq a_f \leq 2 \ \wedge\ v \geq 0 \ \wedge\ v_f \geq 0 \end{aligned}$$

Certificate as a formula

$\exists c_3 \exists c_4 \exists c_5 \exists c_6 \forall v \forall v_f \forall \mathrm{gap} \forall a \forall a_f (\varphi_1 \wedge \varphi_2 \wedge \varphi_3)$

$$\begin{aligned} \text{where} \quad \varphi_1 &\equiv \mathrm{Init} \wedge \mathrm{Inv}' \longrightarrow \mathrm{Inv} \\ \varphi_2 &\equiv \mathrm{Inv} \wedge \mathrm{Inv}' \longrightarrow \mathrm{Safe} \\ \varphi_3 &\equiv p = 0 \wedge \mathrm{Inv}' \longrightarrow \dot{p} \geq 0 \end{aligned}$$

After 1 minute of computation:

- 584 disjuncts, 33365 atomic formulas, depth 13, some still containing $\exists c_5$
- first 33 disjuncts automatically simplify to $c_2^2 - 30c_2 - 75 \leq 0$ for $c_1 > 0$, $c_2 > 0$.
- $\Rightarrow$ no collision for $c_2 = v \leq 32$

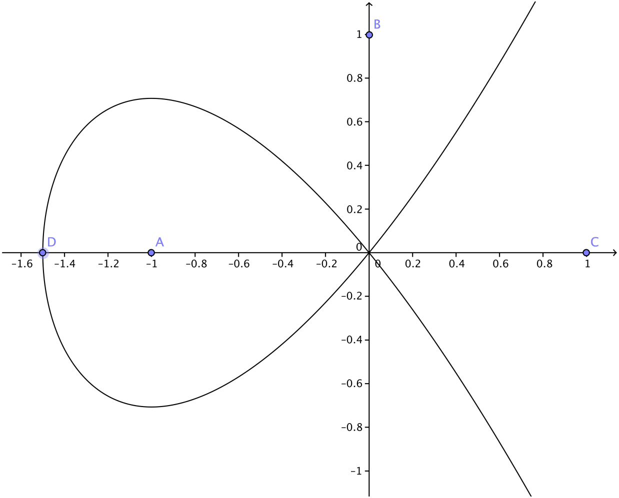
Cylindrical Algebraic Decomposition (CAD)

From Sign Invariant Regions to CAD Cells

$\varphi(f_1, f_2)$ is a Boolean combination of constraints with left hand sides f_1, f_2 and right hand sides 0.

$$f_1(x, y) = 2y^2 - 2x^3 - 3x^2$$

$f_1(A) = -1 < 0$
$f_1(B) = 2 > 0$
$f_1(C) = -5 < 0$
$f_1(D) = 0$



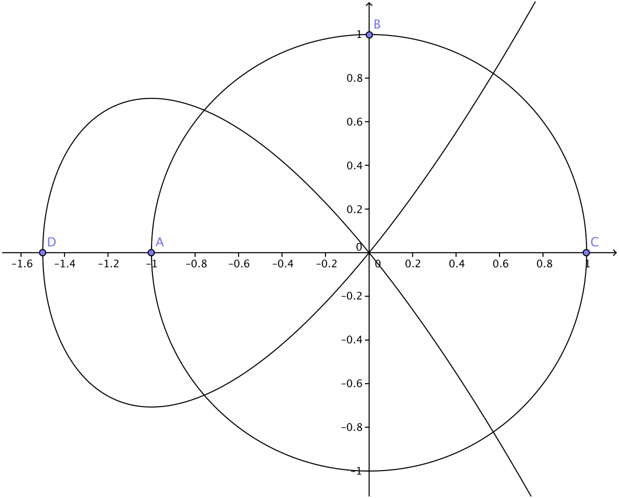
From Sign Invariant Regions to CAD Cells

$\varphi(f_1, f_2)$ is a Boolean combination of constraints with left hand sides f_1, f_2 and right hand sides 0.

$$f_1(x, y) = 2y^2 - 2x^3 - 3x^2$$

$$f_1(A) = -1 < 0$$
$$f_1(B) = 2 > 0$$
$$f_1(C) = -5 < 0$$
$$f_1(D) = 0$$

$$f_2(x, y) = y^2 + x^2 - 1$$



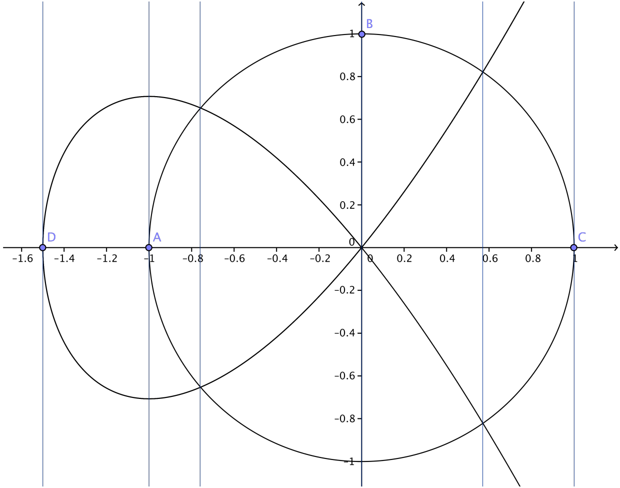
From Sign Invariant Regions to CAD Cells

$\varphi(f_1, f_2)$ is a Boolean combination of constraints with left hand sides f_1, f_2 and right hand sides 0.

$f_1(x, y) = 2y^2 - 2x^3 - 3x^2$

$f_1(A) = -1 < 0$
$f_1(B) = 2 > 0$
$f_1(C) = -5 < 0$
$f_1(D) = 0$

$f_2(x, y) = y^2 + x^2 - 1$



From Sign Invariant Regions to CAD Cells

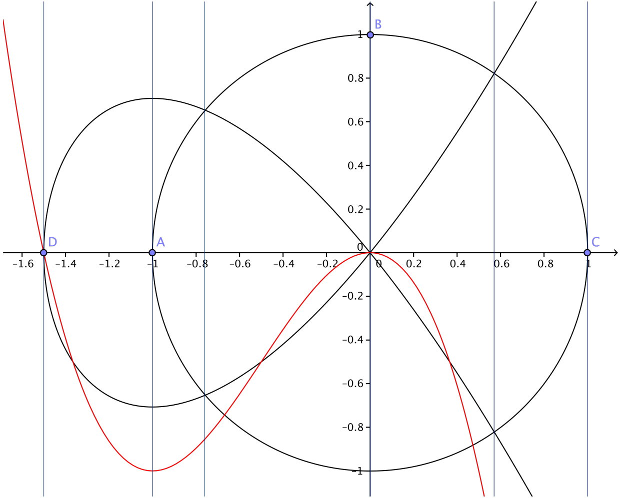
$\varphi(f_1, f_2)$ is a Boolean combination of constraints with left hand sides f_1, f_2 and right hand sides 0.

$f_1(x, y) = 2y^2 - 2x^3 - 3x^2$

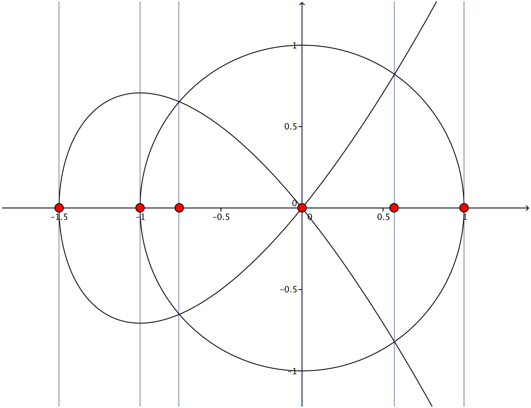
$f_1(A) = -1 < 0$
$f_1(B) = 2 > 0$
$f_1(C) = -5 < 0$
$f_1(D) = 0$

$f_2(x, y) = y^2 + x^2 - 1$

$g(x) = -2x^3 - 3x^2$
...
projection polynomials



Projection and Base Phase (1)

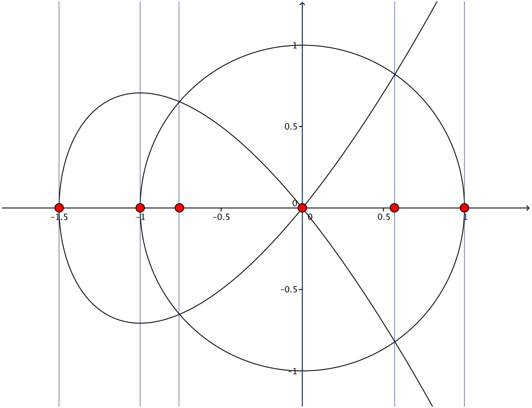


$\varphi(f_1, f_2)$

- **projection operator** computes **projection set**:

$$\Pi(\{f_1(x,y), f_2(x,y)\}) = \{\color{red}{g_1(x), \ldots, g_k(x)}\}$$

Projection and Base Phase (1)



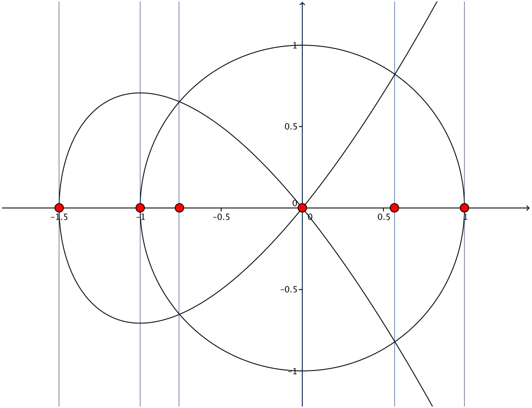
$\varphi(f_1, f_2)$

- **projection operator** computes **projection set**:

$$\Pi(\{f_1(x,y), f_2(x,y)\}) = \{g_1(x), \ldots, g_k(x)\}$$

- Projections of critical points are **among** the zeros of g_1, $\ldots$, g_k.

Projection and Base Phase (1)



$\varphi(f_1, f_2)$

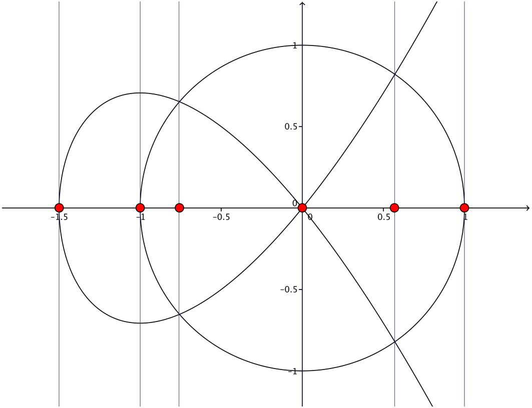
- **projection operator** computes **projection set**:

$$\Pi(\{f_1(x,y), f_2(x,y)\}) = \{g_1(x), \ldots, g_k(x)\}$$

- Projections of critical points are **among** the zeros of $g_1, \ldots, g_k$.
- The zeros of the g_i are real algebraic numbers, e.g.

$$-\sqrt{2} = \left(x^2 - 2,]-10, 1[\right)$$

Projection and Base Phase (1)



$\varphi(f_1, f_2)$

- **projection operator** computes **projection set**:

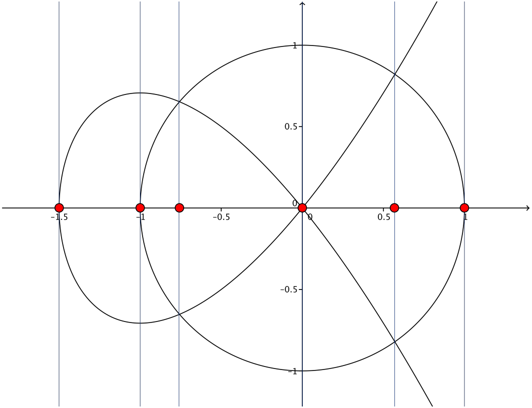
$$\Pi(\{f_1(x,y), f_2(x,y)\}) = \{g_1(x), \ldots, g_k(x)\}$$

- Projections of critical points are **among** the zeros of g_1, $\ldots$, g_k.
- The zeros of the g_i are real algebraic numbers, e.g.

$$-\sqrt{2} = \left(x^2 - 2,]-10, 1[\right)$$

- Their computation is **univariate** computer algebra.

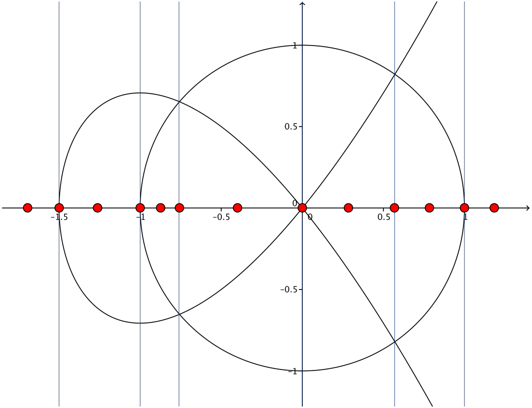
Projection and Base Phase (2)



$\varphi(f_1, f_2)$

- Add points (anywhere) between the zeros as test points for the 1-dimensional cells.

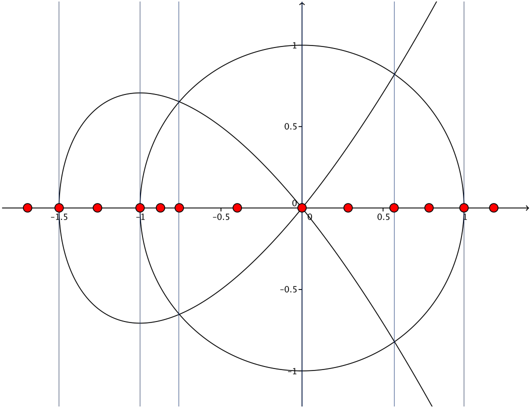
Projection and Base Phase (2)



$\varphi(f_1, f_2)$

- Add points (anywhere) between the zeros as test points for the 1-dimensional cells.
- This yields a **decomposition** of $\mathbb{R}^1$ (the x-axis).

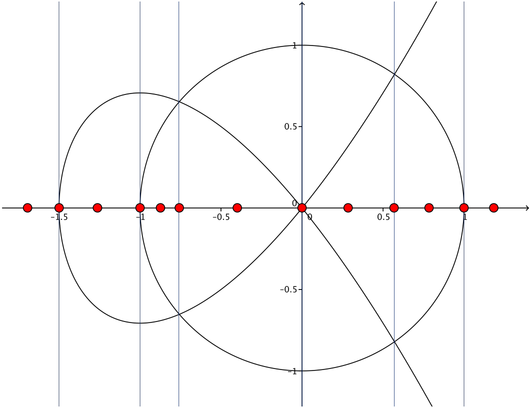
Projection and Base Phase (2)



$\varphi(f_1, f_2)$

- Add points (anywhere) between the zeros as test points for the 1-dimensional cells.
- This yields a **decomposition** of $\mathbb{R}^1$ (the x-axis).
- We want to **lift** this decomposition to $\mathbb{R}^2$.

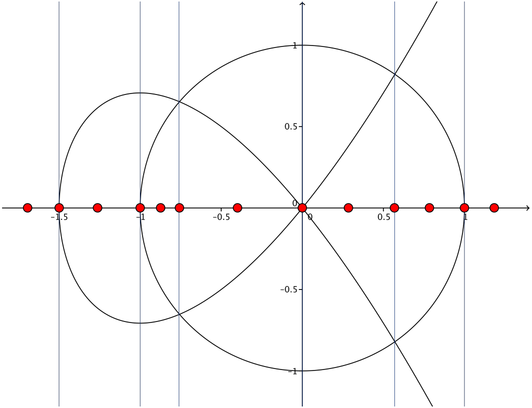
Projection and Base Phase (2)



$\varphi(f_1, f_2)$

- Add points (anywhere) between the zeros as test points for the 1-dimensional cells.
- This yields a **decomposition** of $\mathbb{R}^1$ (the x-axis).
- We want to **lift** this decomposition to $\mathbb{R}^2$.
- By the way: How many cells will there be in $\mathbb{R}^2$?

Extension Phase (Lifting)



$\varphi(f_1, f_2)$

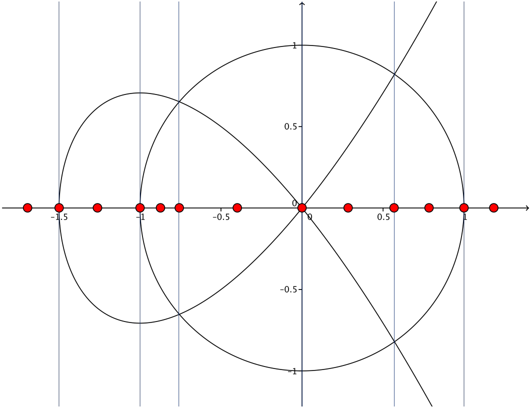
For each test point t from the base phase:

- compute **univariate**

$$f_1(t, y), \quad f_2(t, y).$$

with algebraic number coefficients.

Extension Phase (Lifting)



$\varphi(f_1, f_2)$

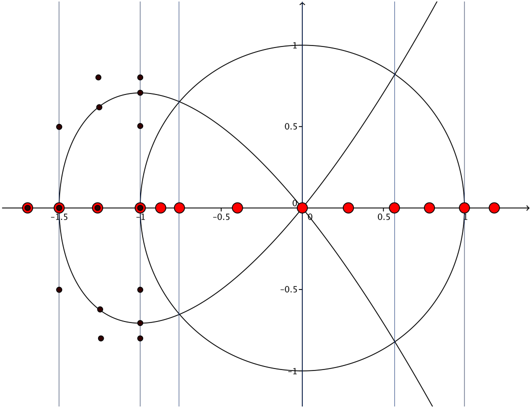
For each test point t from the base phase:

- compute **univariate**

$$f_1(t, y), \quad f_2(t, y).$$

  with algebraic number coefficients.
- compute zeros and points between zeros $u_1, \ldots, u_s$.

Extension Phase (Lifting)



$\varphi(f_1, f_2)$

For each test point t from the base phase:

- compute **univariate**

$$f_1(t, y), \quad f_2(t, y).$$

with algebraic number coefficients.

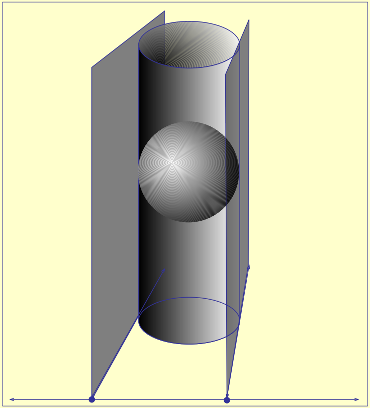
- compute zeros and points between zeros $u_1, \ldots, u_s$.
- this yields test points

$$(t, u_1), \ldots, (t, u_s) \in \mathbb{R}^2$$

for the cylinder over t.

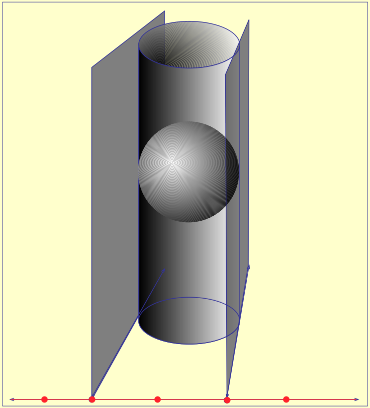
Example: a CAD as a "data structure"

$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$
$P_2 = \{x_2^2 + x_1^2 - 4\}$
$P_1 = \{x_1 + 2, x_1 - 2\}$



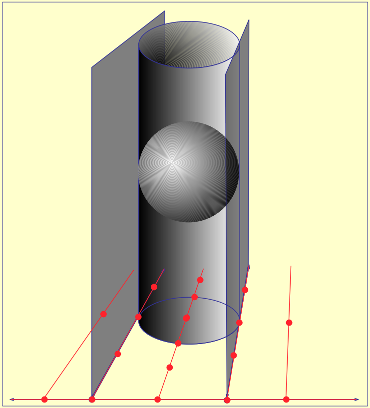
Example: a CAD as a “data structure”

$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



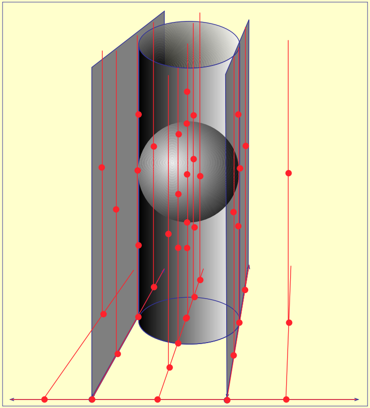
Example: a CAD as a "data structure"

$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$

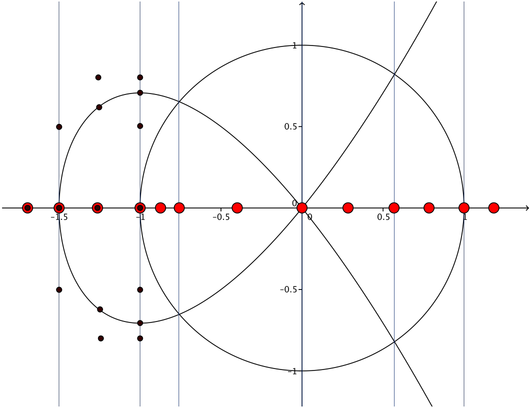


Example: a CAD as a "data structure"

$$P_3 = \{x_1^2 + x_2^2 + x_3^2 - 4\}$$
$$P_2 = \{x_2^2 + x_1^2 - 4\}$$
$$P_1 = \{x_1 + 2, x_1 - 2\}$$



SAT-Checking

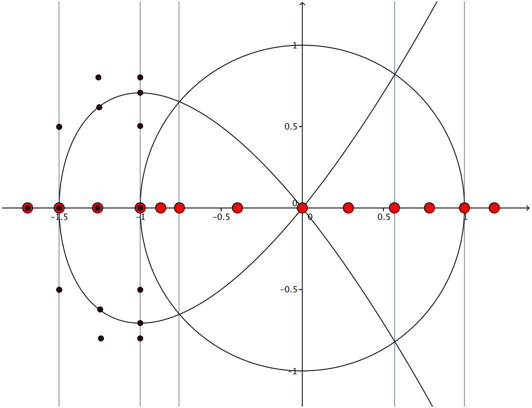


$\varphi(f_1, f_2)$

- Finitely many test points

$$T = \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}), \\ \vdots \\ (t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\}$$

SAT-Checking



$\varphi(f_1, f_2)$

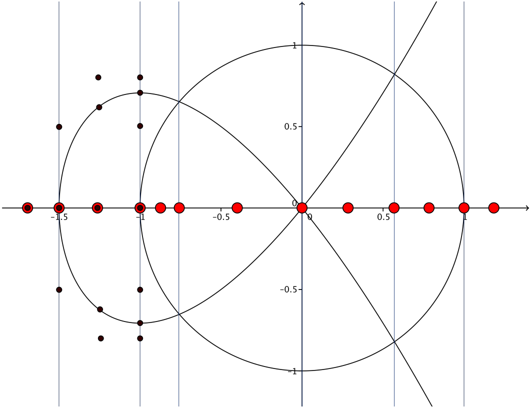
- Finitely many test points

$$T = \begin{array}{l} \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}), \\ \qquad \vdots \\ (t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\} \end{array}$$

- $\mathbb{R} \models \exists\varphi(f_1, f_2)$ iff ex. $t \in T$ s.t.

$$\mathbb{R}, \ (x, y) = t \ \models \ \varphi(f_1, f_2).$$

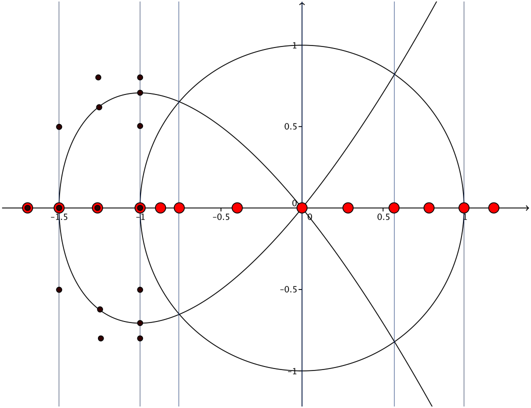
Complete Decision Procedure



- Finitely many test points

$$T = \quad \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}), \\ \vdots \\ (t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\}.$$

Complete Decision Procedure



- Finitely many test points

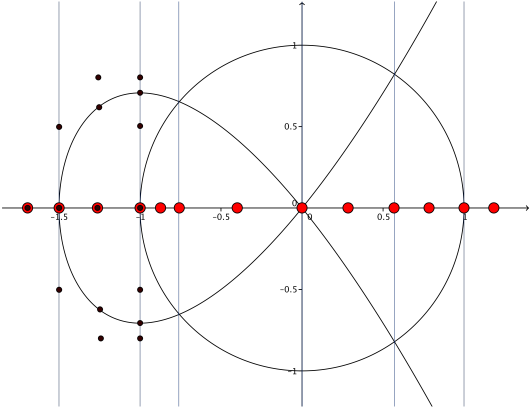
$$T = \begin{array}{l} \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}), \\ \qquad \vdots \\ (t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\}. \end{array}$$

- $\forall x \exists y \varphi(f_1, f_2)$:

  "In each cylinder there is a cell such that ..."

  Satisfying t in each row of T?

Complete Decision Procedure



- Finitely many test points

$$T = \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}),$$
$$\vdots$$
$$(t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\}.$$

- $\forall x \exists y \varphi(f_1, f_2)$:

  "In each cylinder there is a cell such that ..."

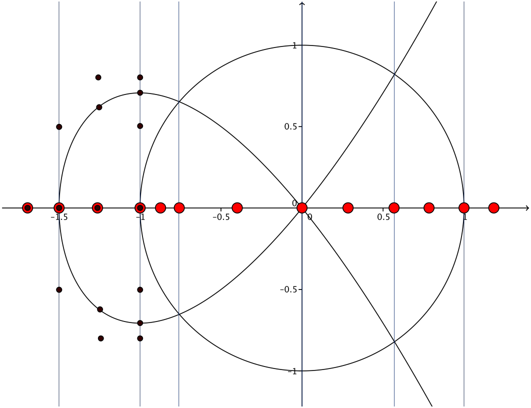
  Satisfying t in each row of T?

- $\exists x \forall y \varphi(f_1, f_2)$:

  "There is a cylinder such that for each cell ..."

  A satisfying column of T?

Complete Decision Procedure



- Finitely many test points

$$T = \quad \{(t_1, u_{t_1,1}), \ldots, (t_1, u_{t_1,s_1}), \\ \vdots \\ (t_r, u_{t_r,1}), \ldots, (t_r, u_{t_r,s_r})\}.$$

- $\forall x \exists y \varphi(f_1, f_2)$:

  "In each cylinder there is a cell such that ..."

  Satisfying t in each row of T?

- $\exists x \forall y \varphi(f_1, f_2)$:

  "There is a cylinder such that for each cell ..."

  A satisfying column of T?

- The innermost variable y was projected first.

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).
- This indicates that the CAD procedure is somewhat an overkill.

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).
- This indicates that the CAD procedure is somewhat an overkill.
- On the other hand, the asymptotic worst complexity $2^{2^{O(n)}}$ in terms of the input word length n is known to be optimal.

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).
- This indicates that the CAD procedure is somewhat an overkill.
- On the other hand, the asymptotic worst complexity $2^{2^{O(n)}}$ in terms of the input word length n is known to be optimal.
- Asymptotically better bounds with refined complexity parameters.

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).
- This indicates that the CAD procedure is somewhat an overkill.
- On the other hand, the asymptotic worst complexity $2^{2^{O(n)}}$ in terms of the input word length n is known to be optimal.
- Asymptotically better bounds with refined complexity parameters.
- In practice, for general input, CAD is the best we have.

Some Remarks Before We Continue

- Given $\varphi(f_1, f_2)$ essentially all the algorithmic work we have done is valid for arbitrary Boolean combinations $\psi(f_1, f_2)$ of arbitrary constraints with left hand sides f_1, f_2 (and right hand sides 0).
- Furthermore, even for arbitrary quantification $QxQ'y$ (in that order).
- This indicates that the CAD procedure is somewhat an overkill.
- On the other hand, the asymptotic worst complexity $2^{2^{O(n)}}$ in terms of the input word length n is known to be optimal.
- Asymptotically better bounds with refined complexity parameters.
- In practice, for general input, CAD is the best we have.
- Until now, we have not used and did not "really know" the cells – only test points.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.

This is relevant also for recent decision procedures (Jovanovic & de Moura).

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1, \ldots, x_k) = Q_{k+1}x_{k+1} \ldots Q_r x_r \varphi(x_1, \ldots, x_k, x_{k+1}, \ldots, x_r)$.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1, \dots, x_k) = Q_{k+1}x_{k+1} \dots Q_r x_r \varphi(x_1, \dots, x_k, x_{k+1}, \dots, x_r)$.
- $x_1, \dots x_k$ are **parameters**.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1, \ldots, x_k) = Q_{k+1}x_{k+1} \ldots Q_r x_r \varphi(x_1, \ldots, x_k, x_{k+1}, \ldots, x_r)$.
- $x_1, \ldots x_k$ are **parameters**.
- Construct CAD with projection order $x_r \to \cdots \to x_{k+1} \to x_k \to \cdots \to x_1$.
  That is, the base phase takes place in $\mathbb{R}[x_1]$.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1, \dots, x_k) = Q_{k+1}x_{k+1} \dots Q_r x_r \varphi(x_1, \dots, x_k, x_{k+1}, \dots, x_r)$.
- $x_1, \dots x_k$ are **parameters**.
- Construct CAD with projection order $x_r \to \dots \to x_{k+1} \to x_k \to \dots \to x_1$.
  That is, the base phase takes place in $\mathbb{R}[x_1]$.
- Consider the finite set $C \subseteq \mathrm{Pot}(\mathbb{R}^k)$ of cells in parameter space,
  i.e., at projection level k with polynomials from $\mathbb{R}[x_1, \dots, x_k]$.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1, \dots, x_k) = Q_{k+1}x_{k+1} \dots Q_r x_r \varphi(x_1, \dots, x_k, x_{k+1}, \dots, x_r)$.
- $x_1, \dots x_k$ are **parameters**.
- Construct CAD with projection order $x_r \to \dots \to x_{k+1} \to x_k \to \dots \to x_1$.
  That is, the base phase takes place in $\mathbb{R}[x_1]$.
- Consider the finite set $C \subseteq \mathrm{Pot}(\mathbb{R}^k)$ of cells in parameter space,
  i.e., at projection level k with polynomials from $\mathbb{R}[x_1, \dots, x_k]$.
- For each $c \in C$ with test point $t_c \in \mathbb{R}^{n-k}$ we can decide $\psi(t_c)$ and collect

$$\mathrm{TRUECELLS} = \{\, c \in C \mid \mathbb{R}, (x_1, \dots, x_k) = t_c \models \psi \,\} \subseteq C.$$

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

- Given $\psi(x_1,\dots,x_k) = Q_{k+1}x_{k+1}\dots Q_r x_r \varphi(x_1,\dots,x_k,x_{k+1},\dots,x_r)$.
- $x_1,\dots x_k$ are **parameters**.
- Construct CAD with projection order $x_r \to \dots \to x_{k+1} \to x_k \to \dots \to x_1$.
  That is, the base phase takes place in $\mathbb{R}[x_1]$.
- Consider the finite set $C \subseteq \mathrm{Pot}(\mathbb{R}^k)$ of cells in parameter space,
  i.e., at projection level k with polynomials from $\mathbb{R}[x_1,\dots,x_k]$.
- For each $c \in C$ with test point $t_c \in \mathbb{R}^{n-k}$ we can decide $\psi(t_c)$ and collect

$$\mathrm{TRUECELLS} = \{\, c \in C \mid \mathbb{R}, (x_1,\dots,x_k) = t_c \models \psi \,\} \subseteq C.$$

- **Assume** that for $c \in C$ we have a quantifier-free **description** formula
  $\Delta_c(x_1,\dots,x_k)$, i.e. $\mathbf{x} \in c$ iff $\mathbb{R} \models \Delta_c(\mathbf{x})$.

Quantifier Elimination

The essential new concept with QE is **quantifier-free description of cells**.
This is relevant also for recent decision procedures (Jovanovic & de Moura).

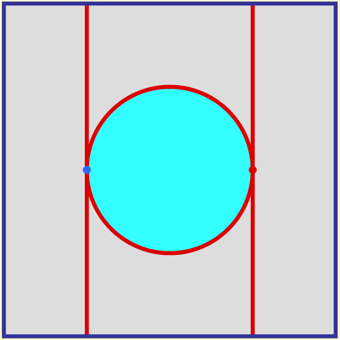
- Given $\psi(x_1, \dots, x_k) = Q_{k+1}x_{k+1} \dots Q_r x_r \varphi(x_1, \dots, x_k, x_{k+1}, \dots, x_r)$.
- $x_1, \dots x_k$ are **parameters**.
- Construct CAD with projection order $x_r \to \dots \to x_{k+1} \to x_k \to \dots \to x_1$.
  That is, the base phase takes place in $\mathbb{R}[x_1]$.
- Consider the finite set $C \subseteq \mathrm{Pot}(\mathbb{R}^k)$ of cells in parameter space,
  i.e., at projection level k with polynomials from $\mathbb{R}[x_1, \dots, x_k]$.
- For each $c \in C$ with test point $t_c \in \mathbb{R}^{n-k}$ we can decide $\psi(t_c)$ and collect

$$\mathrm{TRUECELLS} = \{\, c \in C \mid \mathbb{R}, (x_1, \dots, x_k) = t_c \models \psi \,\} \subseteq C.$$

- **Assume** that for $c \in C$ we have a quantifier-free **description** formula $\Delta_c(x_1, \dots, x_k)$, i.e. $\mathbf{x} \in c$ iff $\mathbb{R} \models \Delta_c(\mathbf{x})$. Then

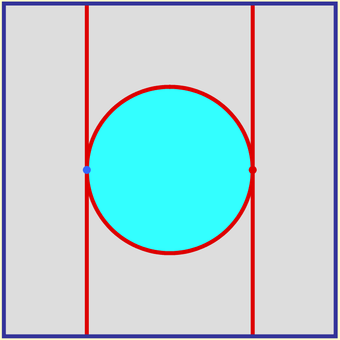
$$\mathbb{R} \models \psi \longleftrightarrow \bigvee_{c \in \mathrm{TRUECELLS}} \Delta_c.$$

Solution Formula Construction Example



$cell$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	T/F
$1,1$	$-$	$-$	$+$	F
$2,1$	0	$-$	$+$	F
$2,2$	0	$-$	0	T
$2,3$	0	$-$	$+$	F
$3,1$	$+$	$-$	$+$	F
$3,2$	$+$	$-$	0	F
$3,3$	$+$	$-$	$-$	T
$3,4$	$+$	$-$	0	F
$3,5$	$+$	$-$	$+$	F
$4,1$	$+$	0	$+$	F
$4,2$	$+$	0	0	F
$4,3$	$+$	0	$+$	F
$5,1$	$+$	$+$	$+$	F

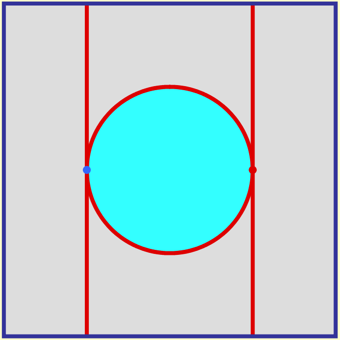
Solution Formula Construction Example



cell	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	T/F
1, 1	−	−	+	F
2, 1	0	−	+	F
2, 2	0	−	0	T
2, 3	0	−	+	F
3, 1	+	−	+	F
3, 2	+	−	0	F
3, 3	+	−	−	T
3, 4	+	−	0	F
3, 5	+	−	+	F
4, 1	+	0	+	F
4, 2	+	0	0	F
4, 3	+	0	+	F
5, 1	+	+	+	F

$$P_{2,1} < 0$$

Solution Formula Construction Example

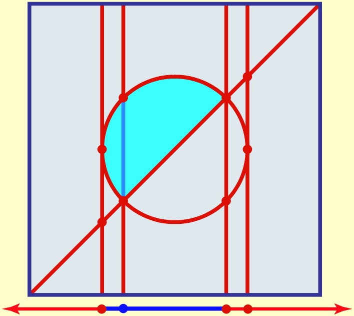


cell	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	T/F
$1,1$	$-$	$-$	$+$	F
$2,1$	0	$-$	$+$	F
$2,2$	0	$-$	0	T
$2,3$	0	$-$	$+$	F
$3,1$	$+$	$-$	$+$	F
$3,2$	$+$	$-$	0	F
$3,3$	$+$	$-$	$-$	T
$3,4$	$+$	$-$	0	F
$3,5$	$+$	$-$	$+$	F
$4,1$	$+$	0	$+$	F
$4,2$	$+$	0	0	F
$4,3$	$+$	0	$+$	F
$5,1$	$+$	$+$	$+$	F

$$P_{2,1} < 0 \;\vee\; P_{1,1} = 0 \wedge P_{2,1} = 0$$

Solution Formula Construction Problem

$$\exists y[x^2 + y^2 - 1 < 0 \ \wedge \ x - y < 0]$$



cell	$x+1$	$x-1$	x^2-2	T/F
1	$-$	$-$	$+$	F
2	0	$-$	$+$	F
3	$+$	$-$	$+$	T
4	$+$	$-$	0	T
5	$+$	$-$	$-$	T
6	$+$	$-$	0	F
7	$+$	$-$	$+$	F
8	$+$	0	$+$	F
9	$+$	$+$	$+$	F

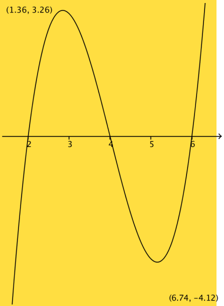
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds “lots of derivatives.”

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.



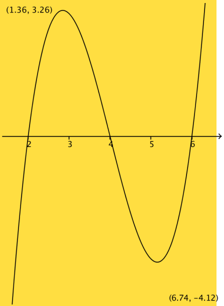
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds "lots of derivatives."

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2, 4[\cup]6, \infty[$, $f = 0$ describes $\{2, 4, 6\}$.



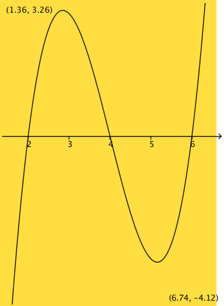
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds “lots of derivatives.”

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2, 4[\cup]6, \infty[$, $f = 0$ describes $\{2, 4, 6\}$.
- f cannot describe exclusively $]2, 4[$ or $\{4\}$.



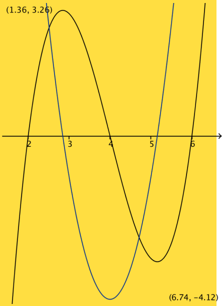
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds “lots of derivatives.”

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2, 4[\cup]6, \infty[$, $f = 0$ describes $\{2, 4, 6\}$.
- f cannot describe exclusively $]2, 4[$ or $\{4\}$.
- $f = 0 \wedge f' = 3x^2 - 24x + 44 < 0$ describes $\{4\}$.

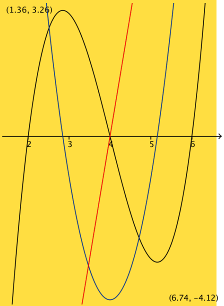


Solutions to the Solution Formula Problem (1)
Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds "lots of derivatives."

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2,4[\cup]6,\infty[$, $f = 0$ describes $\{2,4,6\}$.
- f cannot describe exclusively $]2,4[$ or $\{4\}$.
- $f = 0 \wedge f' = 3x^2 - 24x + 44 < 0$ describes $\{4\}$.
- $f > 0 \wedge f'' = 6x - 24 < 0$ describes $]2,4[$.



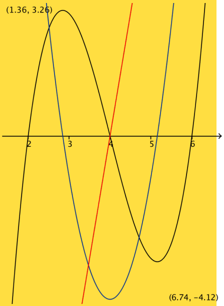
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds “lots of derivatives.”

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2, 4[\cup]6, \infty[$, $f = 0$ describes $\{2, 4, 6\}$.
- f cannot describe exclusively $]2, 4[$ or $\{4\}$.
- $f = 0 \wedge f' = 3x^2 - 24x + 44 < 0$ describes $\{4\}$.
- $f > 0 \wedge f'' = 6x - 24 < 0$ describes $]2, 4[$.
- Isn’t this somehow Rolle’s Theorem? Yes it is!



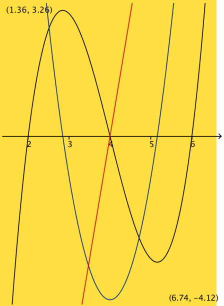
Solutions to the Solution Formula Problem (1)

Augmented Projection

- The approach of the original Collins article (1975).
- Idea: Produce sufficiently many polynomials during projection.
- Technically one adds “lots of derivatives.”

A very simple demonstration of the idea

- Consider a single polynomial $f = x^3 - 12x^2 + 44x - 48$.
- $f > 0$ describes $]2,4[\cup]6,\infty[$, $f = 0$ describes $\{2,4,6\}$.
- f cannot describe exclusively $]2,4[$ or $\{4\}$.
- $f = 0 \wedge f' = 3x^2 - 24x + 44 < 0$ describes $\{4\}$.
- $f > 0 \wedge f'' = 6x - 24 < 0$ describes $]2,4[$.
- Isn't this somehow Rolle's Theorem? Yes it is!



Augmented projection is considered practically infeasible.

Solutions to the Solution Formula Problem (2)

Extended Tarski Language

- PhD thesis of Brown (1999).

Solutions to the Solution Formula Problem (2)

Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_\alpha(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

Solutions to the Solution Formula Problem (2)

Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_\alpha(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

- Predicate is false if f has less than n roots.

Solutions to the Solution Formula Problem (2)

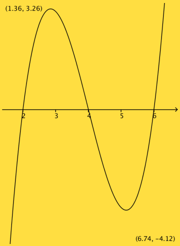
Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_{\alpha}(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

- Predicate is false if f has less than n roots.

Examples



- $f = x^3 - 12x^2 + 44x - 48$ revisited:
  $\operatorname{root}_x(f, 1) < x < \operatorname{root}_x(f, 2)$ describes $]2, 4[$.

Solutions to the Solution Formula Problem (2)

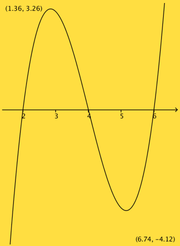
Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_\alpha(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

- Predicate is false if f has less than n roots.

Examples



- $f = x^3 - 12x^2 + 44x - 48$ revisited: $\operatorname{root}_x(f, 1) < x < \operatorname{root}_x(f, 2)$ describes $]2, 4[$.
- In several variables one could obtain, e.g.,

$$\operatorname{root}_\alpha(\alpha^2 - 2, 1) < x < \operatorname{root}_\alpha(\alpha^2 - 2, 2) \wedge$$
$$\operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 3) < y < \operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 5)$$

Solutions to the Solution Formula Problem (2)

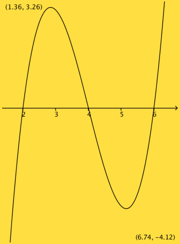
Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_\alpha(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

- Predicate is false if f has less than n roots.

Examples



- $f = x^3 - 12x^2 + 44x - 48$ revisited:
  $\operatorname{root}_x(f, 1) < x < \operatorname{root}_x(f, 2)$ describes $]2, 4[$.
- In several variables one could obtain, e.g.,

$$\operatorname{root}_\alpha(\alpha^2 - 2, 1) < x < \operatorname{root}_\alpha(\alpha^2 - 2, 2) \wedge$$
$$\operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 3) < y < \operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 5)$$

Efficiently check for x, $y \in \mathbb{R}$ if this holds.

Solutions to the Solution Formula Problem (2)

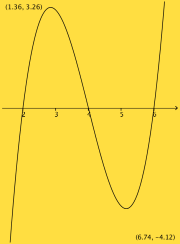
Extended Tarski Language

- PhD thesis of Brown (1999).
- Use extended language with predicates like

$$x \varrho \operatorname{root}_\alpha(f(\alpha), n), \quad \varrho \in \{=, <, >, \leq, \geq, \neq\}.$$

- Predicate is false if f has less than n roots.

Examples



- $f = x^3 - 12x^2 + 44x - 48$ revisited:
  $\operatorname{root}_x(f, 1) < x < \operatorname{root}_x(f, 2)$ describes $]2, 4[$.
- In several variables one could obtain, e.g.,

$$\operatorname{root}_\alpha(\alpha^2 - 2, 1) < x < \operatorname{root}_\alpha(\alpha^2 - 2, 2) \wedge$$
$$\operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 3) < y < \operatorname{root}_\beta(3\beta^7 - \beta + 4x^5, 5)$$

  Efficiently check for $x, y \in \mathbb{R}$ if this holds.

State-of-the-art in QEPCAD and Mathematica, and used in Z3/NLSAT.

Summary

- virtual substitution for real quantifier elimination and some variants
  (extended, generic)
- software: Redlog and other
- other theories
  (integers, comples, differential, padic, terms, queues, PQSAT)
- applications in geometry, verification, ...
- cylindrical algebraic decomposition (CAD)