Machine Learning and Symbolic Computation Part II of II

Matthew England
Coventry University

Thematic Seminar on Symbolic Computation
Shenzhen, China
5 – 12 January 2025

Speaker supported by EPSRC grant EP/T015748/1.

Outline for the Course

Session 1:

1. Introduction
2. Case Study 1:
   ML for Symbolic Integration Algorithm Selection in Maple

Session 2:

3. Case Study 2:
   ML to Optimise Cylindrical Algebraic Decomposition
4. Future Progress from Explainable AI?

Includes joint work with: Rashid Barket, James Bridge, Kelly Cohen, James Davenport, Tereso del Río, Dorian Florescu, Zongyan Huang, Larry Paulson, Lynn Pickering.

Outline

Outline

Motivation: Real QE

Real Quantifier Elimination (Real QE)

Given: A quantified formulae (in prenex normal form) whose atoms are (integral) polynomial constraints;
Produce: a quantifier free formula logically equivalent over $\mathbb{R}$.

Fully quantified examples:

Input: $\exists x, x^2 + 3x + 1 > 0$
Output: $\texttt{True}$ e.g. when $x = 0$
Input: $\forall x, x^2 + 3x + 1 > 0$
Output: $\texttt{False}$ e.g. when $x = -1$
Input: $\forall x, x^2 + 1 > 0$
Output: $\texttt{True}$

Partially quantified example:

Input: $\forall x, x^2 + bx + 1 > 0$
Output: ???

The answer depends on the free (unquantified) variable, b.

Motivation: Real QE

Real Quantifier Elimination (Real QE)

Given: A quantified formulae (in prenex normal form) whose atoms are (integral) polynomial constraints;
Produce: a quantifier free formula logically equivalent over $\mathbb{R}$.

Fully quantified examples:

Input: $\exists x, x^2 + 3x + 1 > 0$
Output: `True` e.g. when $x = 0$
Input: $\forall x, x^2 + 3x + 1 > 0$
Output: `False` e.g. when $x = -1$
Input: $\forall x, x^2 + 1 > 0$
Output: `True`

Partially quantified example:

Input: $\forall x, x^2 + bx + 1 > 0$
Output: ???

The answer depends on the free (unquantified) variable, b.

Partially Quantified QE Example

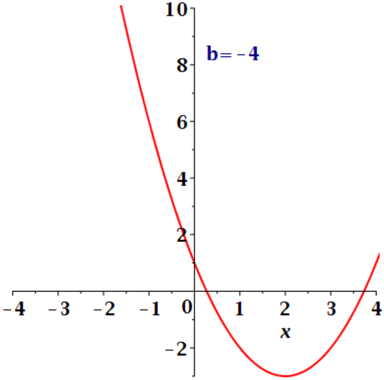
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

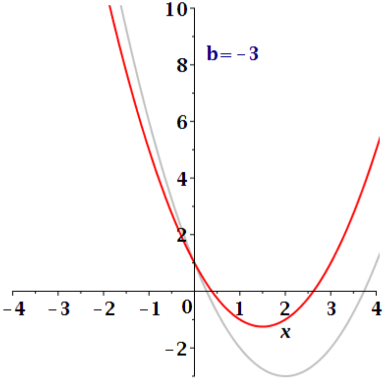
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

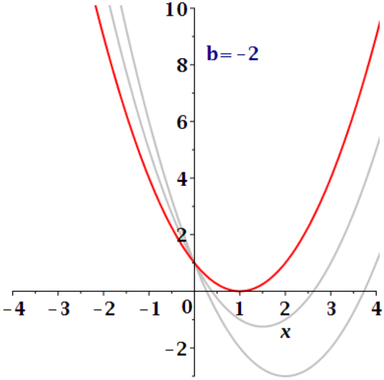
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

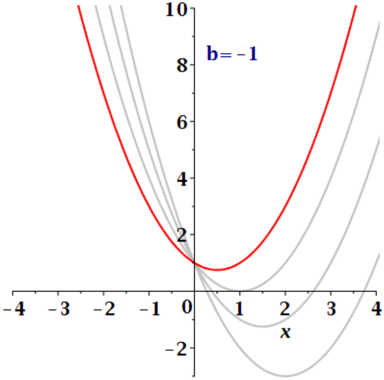
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

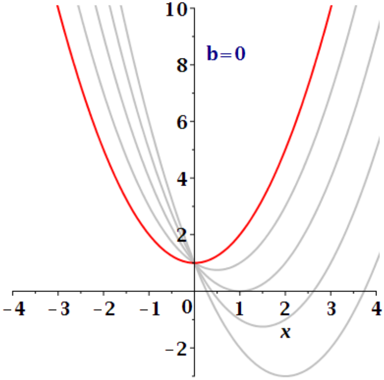
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

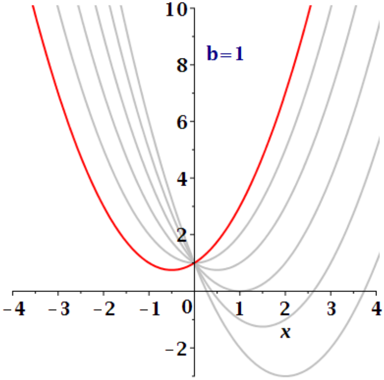
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

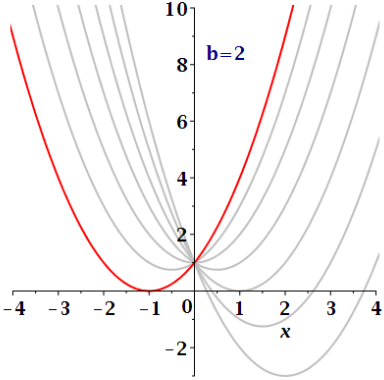
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

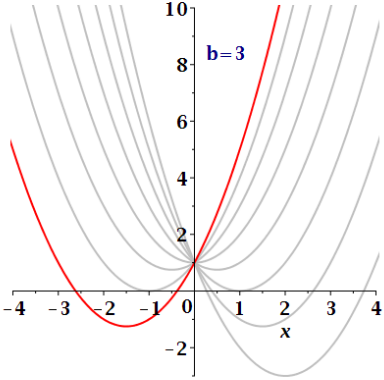
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Partially Quantified QE Example

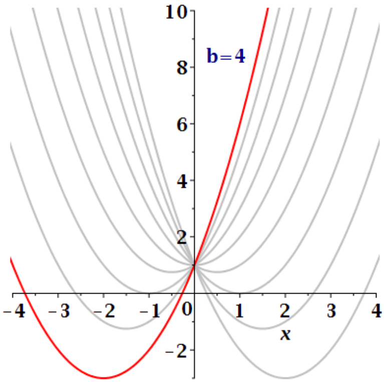
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions algebraically.



Cylindrical Algebraic Decomposition

A **Cylindrical Algebraic Decomposition (CAD)** is:

- a **decomposition** of $\mathbb{R}^n$ into a set of cells C_i (i.e. $\bigcup_i C_i = \mathbb{R}^n$ and $C_i \cap C_j = \emptyset$ if $i \neq j$) such that:
- cells are **semi-algebraic** meaning they may be described by a finite sequence of polynomial constraints;
- cells are arranged **cylindrically**, i.e. projections of any two on a lower coordinate space *in the variable ordering* are identical or disjoint (cells in $\mathbb{R}^m$ stack in cylinders over cells in $\mathbb{R}^{m-1}$).

G.E. Collins.
Quantifier elimination for real closed fields by cylindrical algebraic decomposition.
In Proc. 2nd GI Conf. Automata Theory and Formal Languages, pp. 134–183. Springer-Verlag, 1975.

CAD Invariance Property

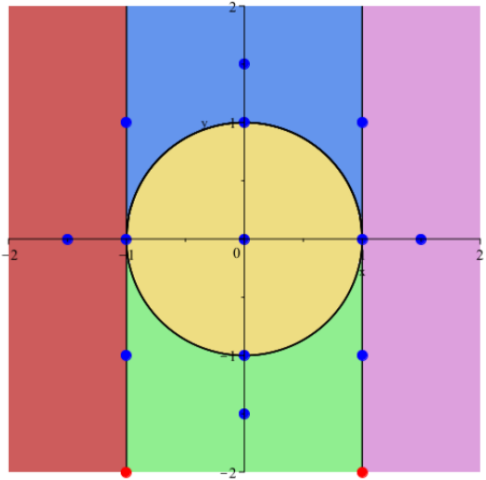
CAD may also refer to an algorithm that produces the CAD object.

The traditional CAD algorithm introduced by Collins in the 1970s takes a set of input polynomials and produces a CAD such that each polynomial has constant sign in each cell: this additional property is called **sign-invariance**.

Such a CAD allows us to uncover properties of polynomials over infinite space by examining a finite set of sample points. In particular, a CAD for the polynomials in a Real QE problem can be used to find the quantifier free equivalent formula.

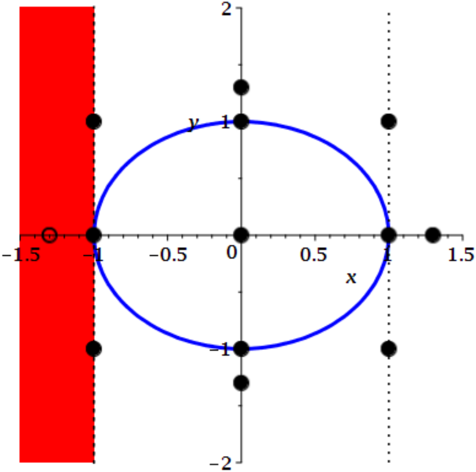
Example: Circle – visualisation (Slide 5/42)

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



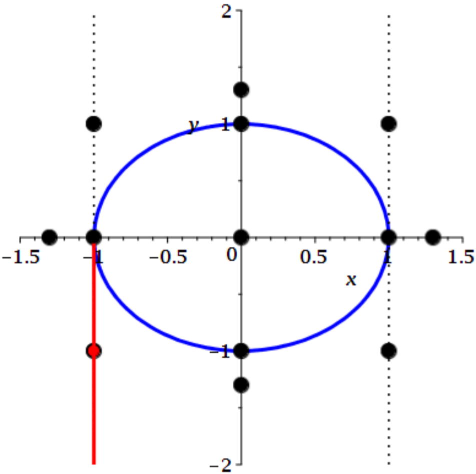
Example: Circle – visualisation

Cell 1: $x < -1,\ y$ free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1,$
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1,\ y$ free



Example: Circle – visualisation

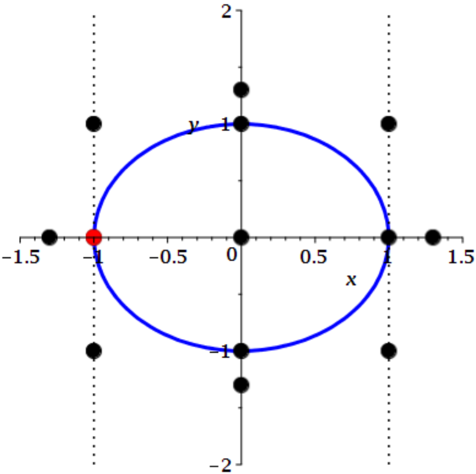
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle – visualisation

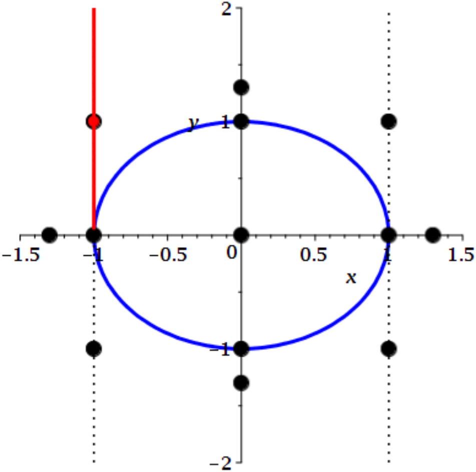
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$

Cell 13: $x > 1$, y free



Example: Circle – visualisation

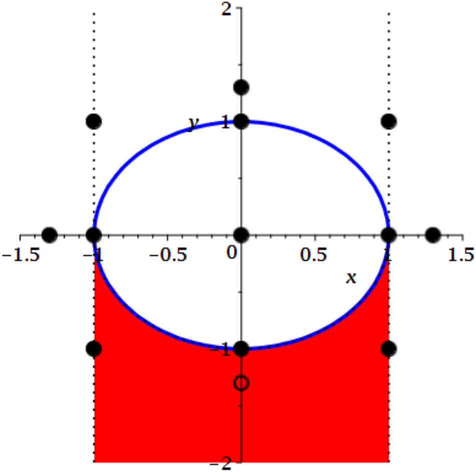
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1,$
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle – visualisation

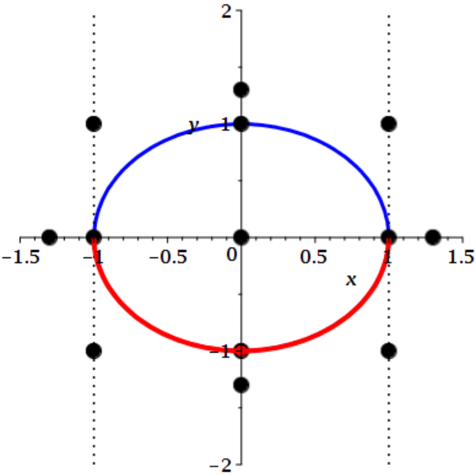
(Slide 5/42)

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1,$
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



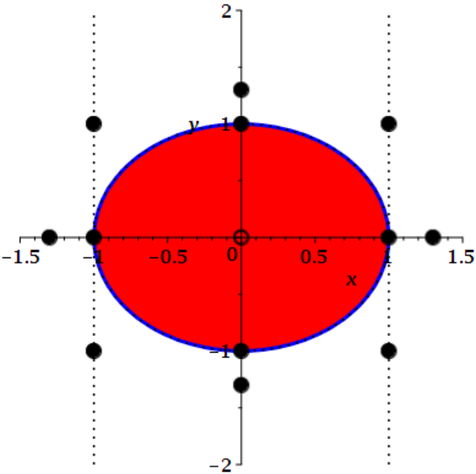
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



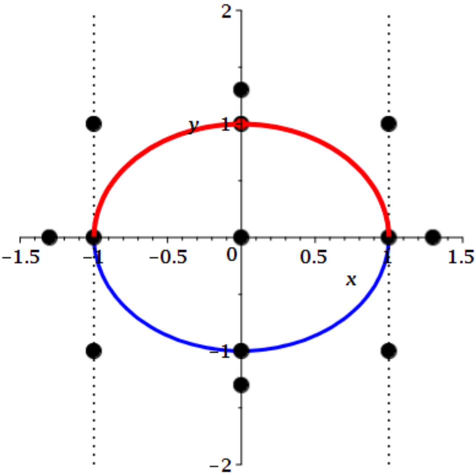
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle — visualisation

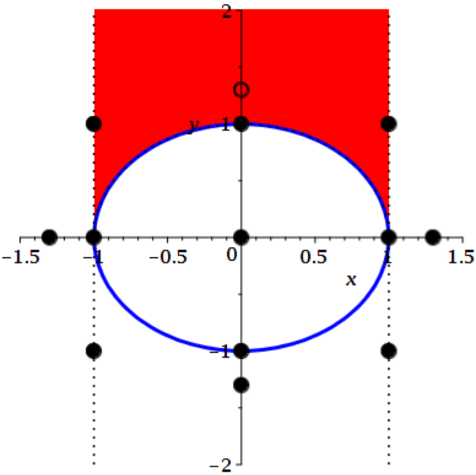
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle – visualisation

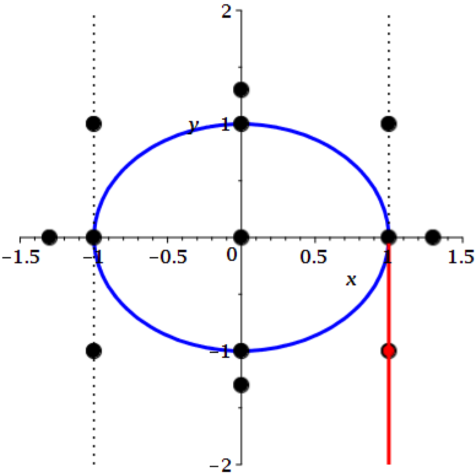
(Slide 5/42)

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



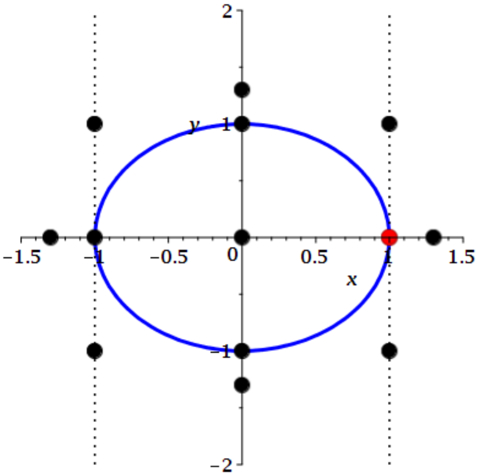
Example: Circle – visualisation (Slide 5/42)

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



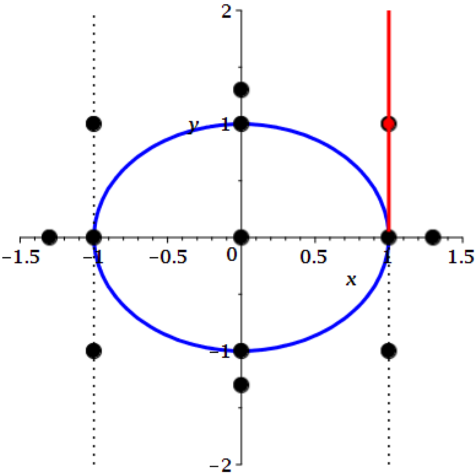
Example: Circle – visualisation (Slide 5/42)

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



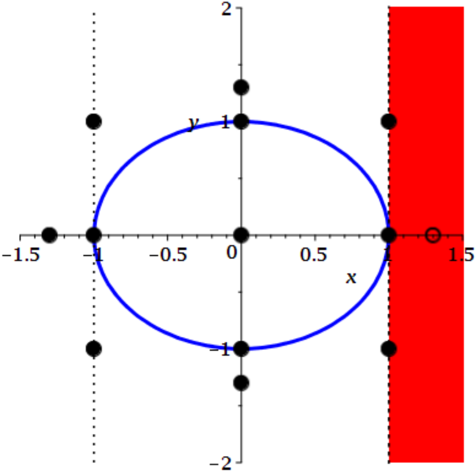
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle – visualisation

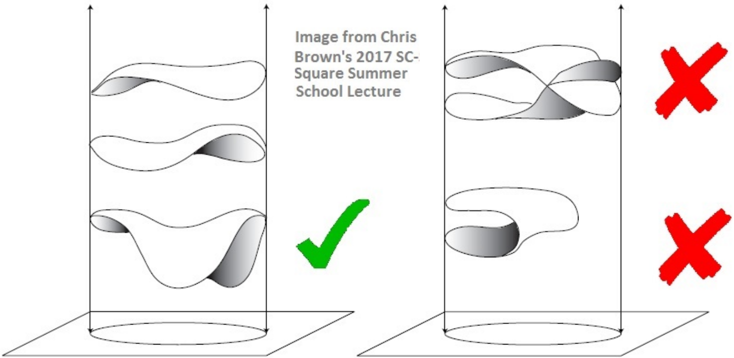
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



How to build a CAD?

The usual approach is to calculate projection polynomials whose roots indicate changes in the behaviour of the input set; decompose in $\mathbb{R}^{n-1}$ with respect to these and then lift each cell to a stack of cells in in $\mathbb{R}^n$ by working at a sample point.

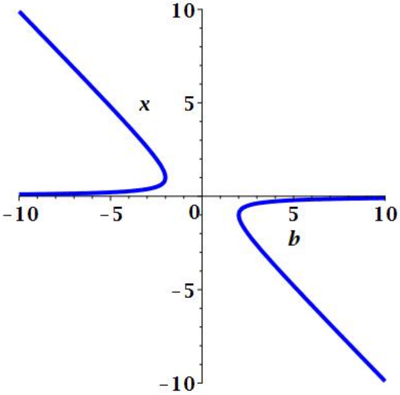
This is a sound approach if the projection provides **delineability**.



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$



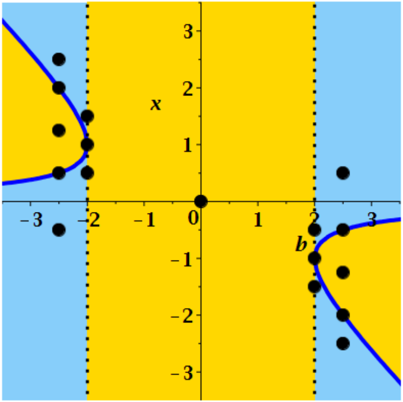
QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.



QE via CAD Example

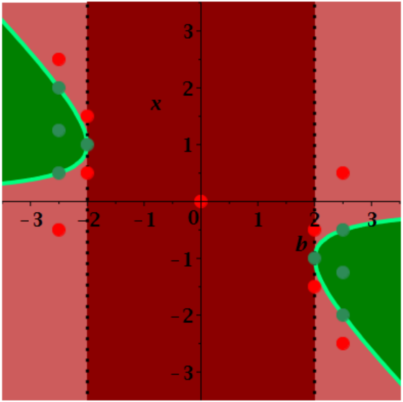
How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

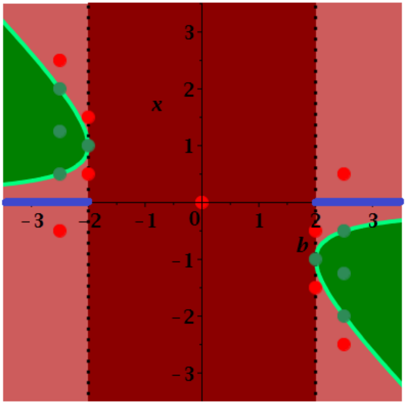
To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.

Take disjunction of projections of true cells:

$$b < -2 \vee b = -2 \vee b = 2 \vee b > 2$$



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

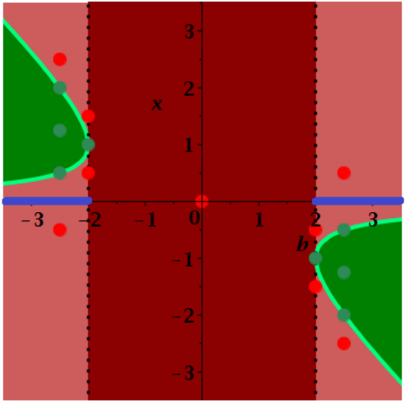
Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.

Take disjunction of projections of true cells:

$$\Longrightarrow$$

$$b \leq -2 \vee b \geq 2$$



QE via CAD in General

In general we can perform Real QE via CAD as follows.

- Eliminate existential quantifiers by projecting the true cells.
- Convert universal quantifiers to existential by using

$$\forall x P(x) = \neg \exists x \neg P(x)$$

and then proceeding with existential and negating the answer.

Recall our original example was $\forall x, x^2 + bx + 1 > 0$. The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$ which from the previous slide we know has solution $b \leq -2 \vee b \geq +2$. The solution to our universally quantified problem is then the negation of this: $-2 < b \wedge b < +2$ (as we found numerically earlier).

Note: Projection and negation are both easy with cylindrical cells!

QE via CAD in General

In general we can perform Real QE via CAD as follows.

- Eliminate existential quantifiers by projecting the true cells.
- Convert universal quantifiers to existential by using

$$\forall x P(x) = \neg \exists x \neg P(x)$$

and then proceeding with existential and negating the answer.

Recall our original example was $\forall x, x^2 + bx + 1 > 0$. The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$ which from the previous slide we know has solution $b \leq -2 \vee b \geq +2$. The solution to our universally quantified problem is then the negation of this: $-2 < b \wedge b < +2$ (as we found numerically earlier).

Note: Projection and negation are both easy with cylindrical cells!

QE via CAD in General

In general we can perform Real QE via CAD as follows.

- Eliminate existential quantifiers by projecting the true cells.
- Convert universal quantifiers to existential by using

$$\forall x P(x) = \neg \exists x \neg P(x)$$

and then proceeding with existential and negating the answer.

Recall our original example was $\forall x, x^2 + bx + 1 > 0$. The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$ which from the previous slide we know has solution $b \leq -2 \vee b \geq +2$. The solution to our universally quantified problem is then the negation of this: $-2 < b \wedge b < +2$ (as we found numerically earlier).

Note: Projection and negation are both easy with cylindrical cells!

QE via CAD Implementations

Both of the big proprietary Computer Algebra Systems have QE implementations: MATHEMATICA (the `Resolve` command) and MAPLE (`RegularChains:-SemiAlgebraicSetTools`; `QuantifierElimination:-CylindricalAlgebraicDecompose`; `RootFinding:-Parametric:-CellDecomposition`); & `SyNRAC`.

The specialist computer algebra system QEPCAD-B is dedicated to QE via CAD and is available for free. It can be used via an intelligent interface TARSKI, is available as a sub-package of SAGEMATH (for Python), and as part of GEOGEBRADISCOVERY which can even run in your browser: `https://autogeo.online/`

CAD implementations also exist in the SMT-solvers, SMT-RAT, YICES2, Z3 and CVC5. All are available for free – but note that these are designed for SMT queries.

Warning: CAD Complexity

By the end of projection you have doubly exponentially many polynomials of doubly exponential degree (in the number of projections, i.e. variables). Hence also the number of real roots, cells and time to compute them grows doubly exponentially!

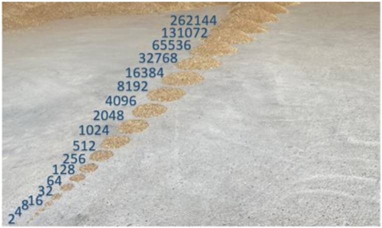
C. Brown and J.H. Davenport.
The complexity of quantifier elimination and cylindrical algebraic decomposition.
In Proc. ISSAC '07, pages 54–60. ACM, 2007.

Thus in practice applications are only realistic for 4 variables, unless specific optimisations are utilised:

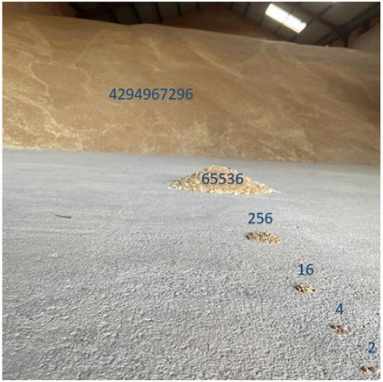
$$2^{2^3} = 256 \qquad 2^{2^4} = 65,536 \qquad 2^{2^5} = 4,294,967,296$$

The Doubly Exponential Wall

Images by Tereso del Rio



Exponential Growth



Doubly Exponential Growth

Alternatives to CAD for QE References (Slide 10/42)

There exists algorithms to achieve Real QE other than via CAD, e.g.:

T. Sturm.
Thirty years of virtual substitution: Foundations, techniques, applications.
In Proc. ISSAC 2018, pages 11–16, ACM, 2018.

R. Fukasaku, H. Iwane, and Y. Sato.
Real quantifier elimination by computation of comprehensive Gröbner systems.
In Proc. ISSAC '15, pages 173–180. ACM, 2015.

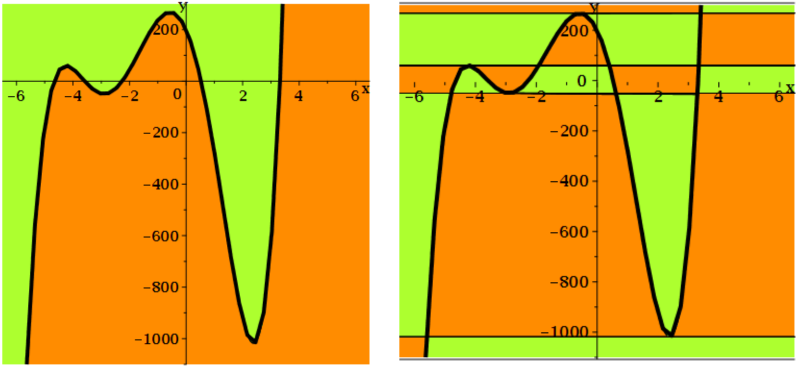
J. Renegar.
Recent progress on the complexity of the decision problem for the reals.
In B.F. Caviness and J.R. Johnson, eds, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 220–241. Springer-Verlag, 1998.

D.Y. Grigor'ev and N.N. Vorobjov.
Solving systems of polynomial inequalities in subexponential time
Journal of Symbolic Computation, 5, pp. 37—64, 1988.

However, CAD remains the general purpose QE backbone, so it is worth thinking what can be done better!

Variable Ordering for CAD

CAD requires a stated variable ordering; which affects decomposition size and computation time.



CAD Variable Ordering Choice

Depending on our application we may have a free or constrained choice in variable ordering. When using CAD for QE we must project variables in the order of quantification, but we are free to change order within quantifier blocks (and with the free variables).

The variable ordering has been long known to effect the number of cells produced in a CAD or even the tractability of a problem.

There are various *human-designed heuristics* to make the choice:

- Simple heuristics which use only simple measures of polynomials (degrees, sparsity etc.)
- Expensive heuristics which use algebraic computations.

CASs tend to apply one of the former; and we focus on those.

More involved heuristics references

A. Dolzmann, A. Seidl, and T. Sturm
Efficient projection orders for CAD.
Proc. ISSAC '04, pp. 111–118, ACM, 2004.
https://doi.org/10.1145/1005285.1005303

D. Wilson, M. England, R. Bradford and J.H. Davenport.
Using the distribution of cells by dimension in a cylindrical algebraic decomposition.
Proc. SYNASC '14, pp. 53–60, IEEE, 2014.
http://dx.doi.org/10.1109/SYNASC.2014.15

H. Li and B. Xia and H. Zhang and T. Zheng.
Choosing better variable orderings for cylindrical algebraic decomposition via exploiting chordal structure
Journal of Symbolic Computation, 116, pp. 324–344, 2023.
https://doi.org/10.1016/j.jsc.2022.10.009

Simple human-designed heuristics

Brown's Heuristic: chooses variable based on overall degree; breaking ties with total degree of the terms with that variable; breaking ties with number of terms the variable is in.

C. Brown
Companion to the Tutorial: Cylindrical Algebraic Decomposition, presented at ISSAC 2004.
www.usna.edu/Users/cs/wcbrown/research/ISSAC04/handout.pdf

gmods: chooses variable based on the sum of its degree in each polynomial.

T. del Río and M. England
New Heuristic to Choose a Cylindrical Algebraic Decomposition Variable Ordering Motivated by Complexity Analysis
Proc. CASC 2022, LNCS 13366, pp. 300––317. Springer, 2022.
https://doi.org/10.1007/978-3-031-14788-3_17

Improved CAD References

(Slide 15/42)

G.E. Collins and H. Hong.
Partial cylindrical algebraic decomposition for quantifier elimination
Journal of Symbolic Computation, 12, pp. 299–328, 1991.
https://doi.org/10.1016/S0747-7171(08)80152-6

C. Brown and S. McCallum.
Enhancements to Lazard's Method for Cylindrical Algebraic Decomposition
Proc. CASC '20, LNCS 12291, pp. 129–149, 2020.
https://doi.org/10.1007/978-3-030-60026-6_8

A. Strzeboński.
Cylindrical Algebraic Decomposition using validated numerics
Journal of Symbolic Computation, 41:9, pp. 1021–1038, 2006.
https://doi.org/10.1016/j.jsc.2006.06.004

C. Chen and M. Moreno Maza and B. Xia and L. Yang.
Computing cylindrical algebraic decomposition via triangular decomposition
Proc. ISSAC '09, pp. 95–102, 2009.
https://doi.org/10.1145/1576702.1576718

Rizeng Chen.
A Geometric Approach to Cylindrical Algebraic Decomposition.
Preprint, arXiv:2311.10515 (Oct 2024). https://arxiv.org/abs/2311.10515

All still have the variable ordering choice.

Reformulated CAD Theory References (Slide 16/42)

We can reformulate how CAD theory is turned into algorithms:

D. Jovanovic and L. de Moura
Solving Non-linear Arithmetic
Proc. IJCAR '12, LNCS 7364, pp. 339–354, 2012.
https://doi.org/10.1007/978-3-642-31365-3_27

C. W. Brown
Open Non-uniform Cylindrical Algebraic Decompositions
Proc. ISSAC 2015, pp. 85–92. ACM, 2015
https://doi.org/10.1145/2755996.2756654

E. Ábrahám, J.H. Davenport, M. England and G. Kremer.
Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings.
JLAMP 119, pages 2352-2208. Elsevier, 2021.
https://doi.org/10.1016/j.jlamp.2020.100633

G. Kremer and J. Nalbach.
Cylindrical Algebraic Coverings for Quantifiers.
Proc. SC^2 2022, CEUR-WS 3458, pp. 1–9, 2023.
https://ceur-ws.org/Vol-3458

All still have the variable ordering choice.

Outline

First Attempt of ML for CAD variable ordering (Slide 17/42)

First attempt used a support vector machine classifier to choose which of the three variable ordering heuristics we had implemented in our CAD code to follow for a given problem instance: no one heuristic dominated the others; ML choice did better than any one.

Z. Huang, M. England, D. Wilson, J. Davenport, L. Paulson and J. Bridge.
Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition.
Intelligent Computer Mathematics (LNCS 8543), pp. 92-107. Springer Berlin Heidelberg, 2014. http://dx.doi.org/10.1007/978-3-319-08434-3_8

Actually, the first use of Machine Learning to optimise computer algebra.

EPSRC ML4QE Project 2018–2020

Experimented with ML methodology to choose the ordering:

- Classifier model choice.
- New feature extraction methods to represent polynomials to Machine Learning models: brute force combinations of simple statistics followed by selection.
- Tailored hyper-parameter selection to runtime rather than classification accuracy.

D. Florescu and M. England.
A Machine Learning Based Software Pipeline to Pick the Variable Ordering for Algorithms with Polynomial Inputs.
Mathematical Software (LNCS 12097), pp. 302–311. Springer International Publishing, 2020.
https://doi.org/10.1007/978-3-030-52200-1_30

Dataset Issues

Above work used the `QF_NRA` benchmarks from the SMT-LIB: a collection of satisfiability problems whose atoms are non-linear polynomial constraints over the reals.

Chosen as the largest (only substantial) existing set of benchmarks for CAD. Problems are meaningful: come from "*industry*" (theorem provers MetiTarski and Keymaera), biology, economics, dynamic geometry proofs, etc. **However:**

1. Dataset is highly imbalanced with respect to CAD variable ordering. Risk of over-fitting given no reason to expect imbalance in general data.
2. Data is highly dominated by one source of problems and these consist of many problems that differ only by a constant. Risk of data leakage between testing and training.

Dataset Augmentation to Solve Issue 1

Use variable permutations on problem instances to:

1. Balance dataset: random permutation on each instance.
2. Augment dataset: from each problem instance create a set of instances from all possible permutations.

Tereso del Río and M. England.
Data Augmentation for Mathematical Objects.
Proc. SC^2 2023, CEUR-WS 3455, pp. 29––38, 2023.
https://ceur-ws.org/Vol-3455/

Hester, Hitaj, Passmore, Owre, Shankar, and Yeh.
An Augmented MetiTarski Dataset for Real Quantifier Elimination Using Machine Learning.
Proc. CICM 2023, LNCS 14101, pp. 297––302, 2023.
https://doi.org/10.1007/978-3-031-42753-4_21

The loss in "*performance*" observed from moving to a balanced test set is more than made up for by using an augmented training set.

Dataset Augmentation Results

Testing dataset	Unbalanced	Balanced	Augmented
KNN-Unbalanced	0.51	0.21	0.26
DT-Unbalanced	0.53	0.31	0.31
SVC-Unbalanced	0.48	0.23	0.2
RF-Unbalanced	**0.58**	**0.35**	**0.37**
MLP-Unbalanced	0.51	0.32	0.32

Accuracy of models trained on the unbalanced dataset, when tested on the different testing datasets.

Testing dataset	Unbalanced	Balanced	Augmented
KNN-Balanced	0.41	0.36	0.38
DT-Balanced	0.43	0.45	0.45
SVC-Balanced	0.25	0.3	0.28
RF-Balanced	**0.49**	**0.52**	**0.52**
MLP-Balanced	0.45	0.43	0.44

Accuracy of models trained on the balanced dataset, when tested on the different testing datasets.

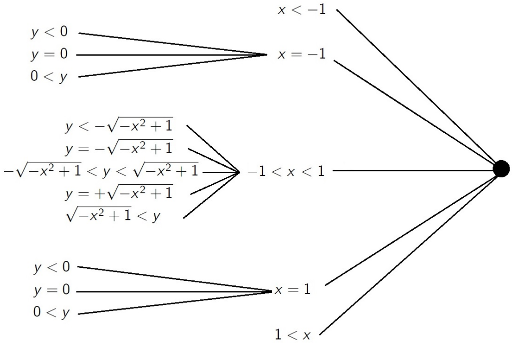
Testing dataset	Unbalanced	Balanced	Augmented
KNN-Augmented	0.54	0.55	0.53
DT-Augmented	0.54	0.55	0.54
SVC-Augmented	0.46	0.48	0.49
RF-Augmented	**0.62**	**0.63**	**0.61**
MLP-Augmented	0.48	0.5	0.51

Accuracy of models trained on the augmented dataset, when tested on the different testing datasets.

Merging Problems to Solve Issue 2

(Slide 22/42)

We merge problem instances who have the same CAD tree structure in each variable ordering.



These has no meaningful loss in learning ability, despite meaning a much smaller dataset (about one sixth the size)!

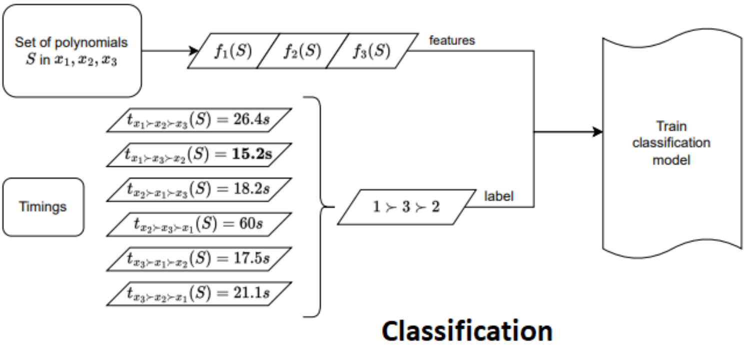
From Classification to Regression

The above framed this as ML *Classification* (choose from discrete set of variable orderings). May be reframed as ML *Regression* predict the time taken with a particular ordering: multiple predictions then compared to choose the final ordering.

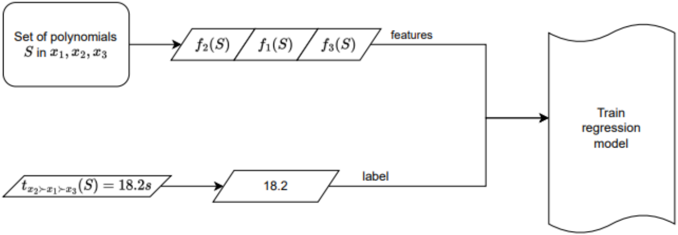
- A model trained with regression has access to more information: not just which ordering did best but also which came second, which third etc.
- However, regression is a more difficult task (but we only need to be good enough at it to make a ranking).

T. del Río and M. England.
Lessons on Datasets and Paradigms in Machine Learning for Symbolic Computation: A Case Study on CAD..
Mathematics in Computer Science, 18, article 17, Springer, 2024.
Preprint: https://doi.org/10.48550/arXiv.2401.13343

Classification vs Regression Flowcharts

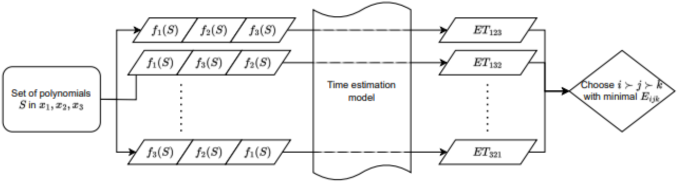


Classification vs. Regression Flowcharts



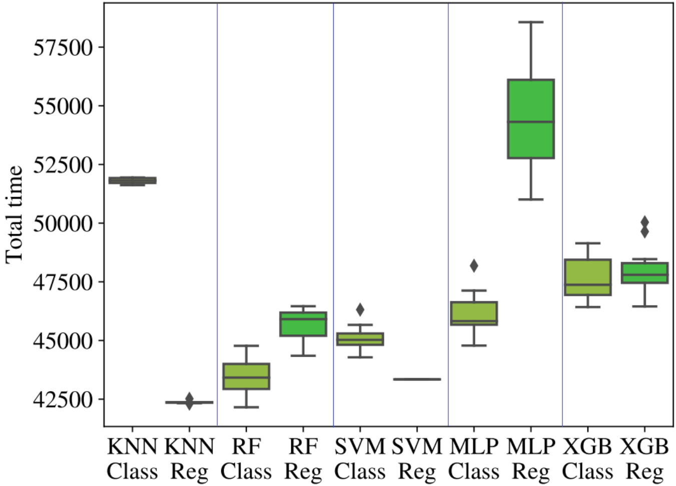
Regression Training

Regression Testing



Classification vs. Regression Results

Regression sometimes but not always beneficial.



Recent work by others with GNNs and RL

Jia, Dong, Liu, Huang, Ma, Zhang.
Suggesting Variable Order for Cylindrical Algebraic Decomposition via Reinforcement Learning.
Proc. NIPS 2023. https://openreview.net/forum?id=vNsdFwjPtL

Uses Graph Neural Networks (nodes being variables and edges if two variables appear in the same polynomial).

Experiments with reinforcement learning instead of supervised learning to train an agent to make the decision.

Trained on random data: performs very well on random data, but less well on SMT-LIB data (still outperformed by `gmods`).

Generalisation Problems of ML for CAD

All the ML approaches presented to date suffer a drop-off in performance when models are applied to alternative datasets.

- Training on real world data is limited to large datasets (even with augmentation). The large datasets available are not representative of all problem domains. E.g. important applications in biology and economics but datasets only number in the 100s of instances.
- Training on random data does not generalise well to non-random data: the real world is not "*random*".

Is ML optimisation of CAD limited to domain specific use?

Outline

Inspiration

D. Peifer, M. Stillman, and D. Halpern-Leistner.
Learning selection strategies in Buchberger's algorithm,
In: Proc. ICML 2020, pages 7575-7585, PMLR, 2020.

Applied reinforcement learning to choose the order in which to process S-pairs in Buchberger's algorithm for a Gröbner Basis.

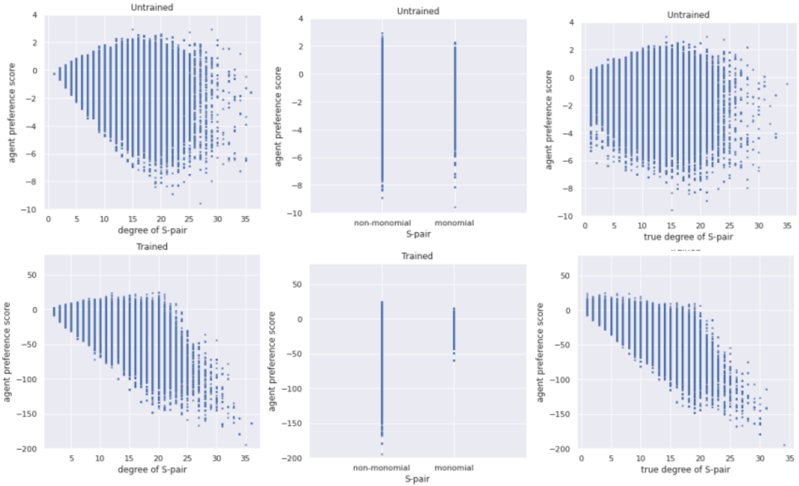
Human analysis of their model revealed that it seemed to use some simple strategies but novel strategies:

- Prefer pairs with low lcm of LMs (Degree strategy).
- Prefer pairs whose S-polynomials are monomials.
- Prefer pairs whose S-polynomials are low degree (true degree).

Suggests that ML can reveal new mathematical truths about the algorithm, and direct future non-ML algorithm development.

How to automate such analysis?

From Peifer's Thesis (2021)



Inspiration

D. Peifer, M. Stillman, and D. Halpern-Leistner.
Learning selection strategies in Buchberger's algorithm,
In: Proc. ICML 2020, pages 7575-7585, PMLR, 2020.

Applied reinforcement learning to choose the order in which to process S-pairs in Buchberger's algorithm for a Gröbner Basis.

Human analysis of their model revealed that it seemed to use some simple strategies but novel strategies:

- Prefer pairs with low lcm of LMs (Degree strategy).
- Prefer pairs whose S-polynomials are monomials.
- Prefer pairs whose S-polynomials are low degree (true degree).

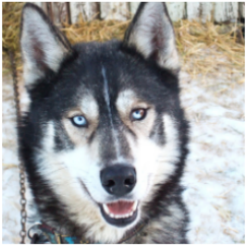
Suggests that ML can reveal new mathematical truths about the algorithm, and direct future non-ML algorithm development.

How to automate such analysis?

Explainable AI (XAI)

Explainable AI (XAI) refers to AI systems that can provide explanations for their actions. useful:

- As an error check on the ML process.
- To build trust in the systems; regulatory compliance.



(a) Husky classified as wolf

(b) Explanation

Idea: XAI applied to ML models guide CAS development?

XAI Taxonomy

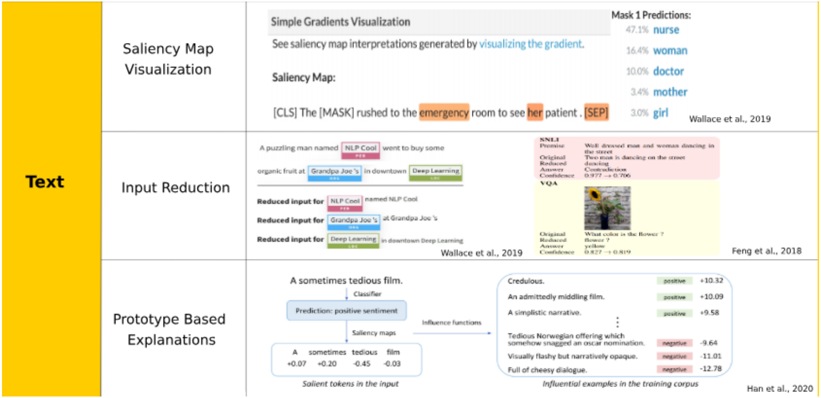
Local XAI seeks to explain an individual prediction/decision.
Global XAI seeks to explain the decision procedure. We are more interested in global explanations.

Two main families of XAI methods:

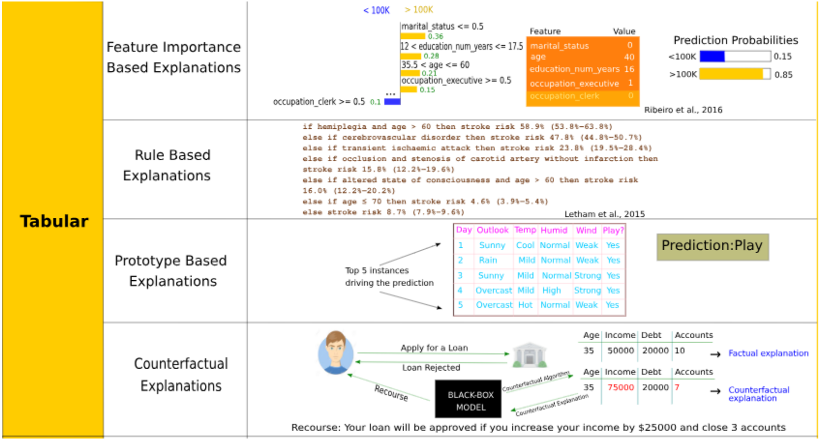
- **Ante-hoc explainable models** are transparent by design in their decisions, e.g. linear regression and support vector machines. Such methods are sometimes called **Interpretable ML**. Perceived trade-off between interpretability and accuracy.
- **Post-hoc explainability** is where an ML model is trained as normal, and then a secondary analysis provides an explanation for why the decision was made (the secondary analysis may have its own inaccuracies independent of the original ML).

Both methods worth exploring.

From Maveli 2021 (Spectra) Text



From Maveli 2021 (Spectra) Tabular



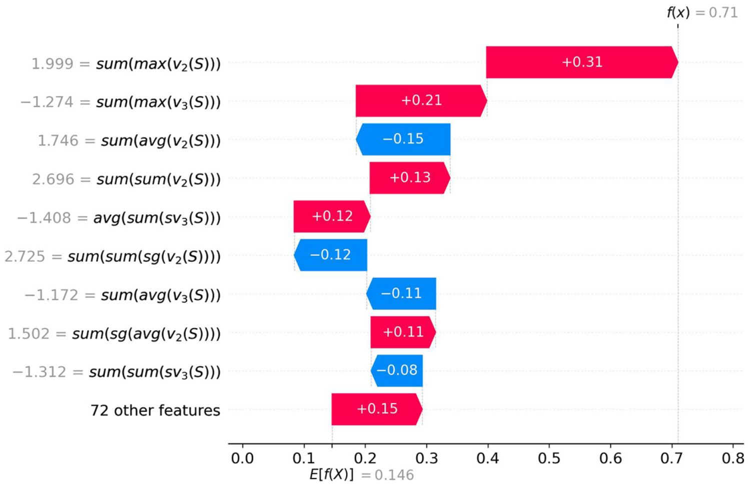
Outline

Post-hoc XAI for CAD Variable Ordering

We applied the SHAP XAI tool to analyse the features used by ML classifiers to choose the CAD variable ordering.

L. Pickering, T. del Rıio Almajano, M. England and K. Cohen.
Explainable AI Insights for Symbolic Computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition.
Journal of Symbolic Computation, **123**, Article Number 102276, 2024.

Example SHAP Waterfall Plot



What to do with the SHAP output?

- We aggregated the feature importance scores across the dataset for each of our ML models (KNN, MLP, SVM, RF).

This allowed us to see which models used which features – they did not fully agree on feature ranking.

- We had the models "*vote*" (used the Dowdall System) on features to create an overall feature ranking.

Some features identified as impactful had been known before; others were new.

- Took the top 6 voted features and studied all 120 possible ordered triples of them to create variants with the structure of Brown's heuristic.

The very best of these were the top three ranked features in that order!

What to do with the SHAP output?

- We aggregated the feature importance scores across the dataset for each of our ML models (KNN, MLP, SVM, RF).

This allowed us to see which models used which features – they did not fully agree on feature ranking.

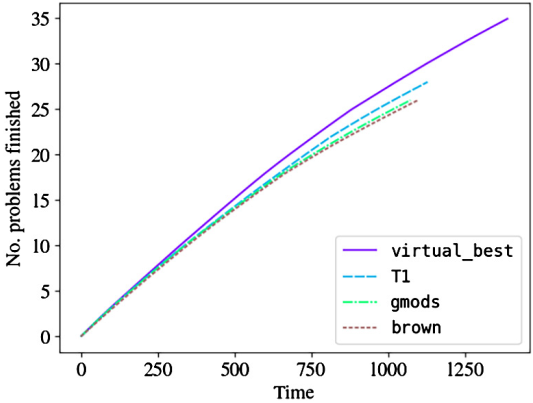
- We had the models "*vote*" (used the Dowdall System) on features to create an overall feature ranking.

Some features identified as impactful had been known before; others were new.

- Took the top 6 voted features and studied all 120 possible ordered triples of them to create variants with the structure of Brown's heuristic.

The very best of these were the top three ranked features in that order!

Performance of the best triple on SMT-LIB



We picked the structure of Brown's heuristic to compare to something known.

There is clearly potential to do even better by explore more nuanced structures.

Ante-hoc XAI for CAD Variable Ordering

The SHAP analysis was post-hoc XAI. Can we use something inherently interpretable?

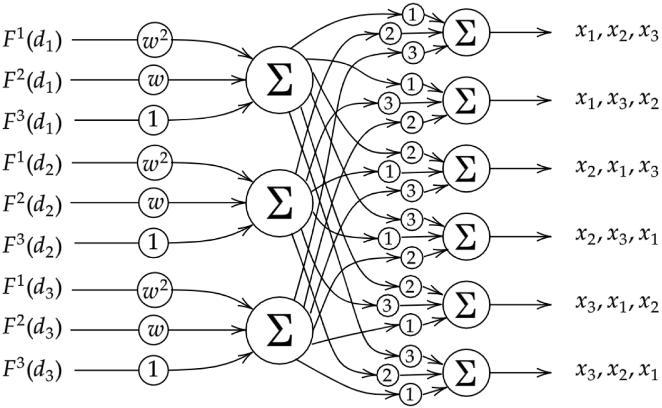
Our original ML work used fairly simple ML models (SVMs): although considered more interpretable than deep learning they did not offer any easy interpretations to us!

We considered trying to design an interpretable model by, once again, taking inspiration from the Brown heuristic.

D. Florescu and M. England.
Constrained Neural Networks for Interpretable Heuristic Creation to Optimise Computer Algebra Systems.
Proc. ICMS 2024 (LNCS 14749), pp. 186-195,
https://doi.org/10.1007/978-3-031-64529-7_19

Brown Heuristic as Neural Network (Slide 38/42)

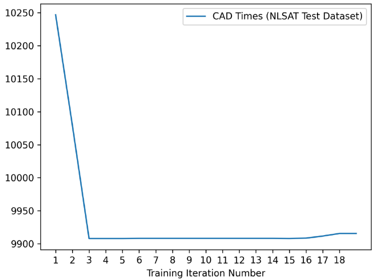
Started by formulating Brown's heuristic as a neural network:



Where w is chosen so that $F^i(d_v) < 2 - 1$ for all features and variables.

Searching the space of similar networks

Then starting from the "*Brown Neural Network*" we optimise the weights by ML training.
We obtain something similar to Brown's heuristic (now linear combinations at each stage).



Dataset Note: In this experiment we trained on polynomials whose exponents and coefficients were created randomly from distributions with the same mean and standard deviation as the SMT-LIB, while testing on the actual SMT-LIB data.

Generalisation over datasets!

Both experiments use XAI techniques to produce heuristics for CAD variable ordering and both have been shown to produce generalisability over datasets:

- In [FE24] we actually trained on random data and and tested on SMTLIB data.
- In [PdREC24] we trained and tested on the SMT-LIB. But subsequent experiments below on very large amounts of random data showed generalisability.

C. Chen, R.-J. Jing, C. Qian, Y. Yuan, and Y. Zhao.
A dataset for suggesting variable orderings for cylindrical algebraic decompositions.
In Proc. CASC 2024, (LNCS 14938), 100–119. Springer, 2024. https://doi.org/10.1007/978-3-031-69070-9_7

Summary of our Two XAI Experiments

Both experiments took a *human-designed* heuristic as a starting point and then used ML processes to produce another heuristic.

The outputs are not human-designed but are *human-level*:

- can be expressed in natural language in a similar amount of text as a heuristic designed by a human;
- can be implemented without any ML architecture.

They could thus be employed by any CAD developer without having to do any ML. Further, they perform almost as well as the ML models themselves and generalise better.

We suggest that these could exemplify a new general methodology for computer algebra heuristic design.

Thanks for Listening

Matthew.England@coventry.ac.uk
https://matthewengland.coventry.domains