Real Quantifier Elimination: Recent Algorithmic Progress and Applications

Matthew England
Coventry University
Centre for Computational Science and Mathematical Modelling

British Computer Society Coventry Branch
15th November 2023

Speaker supported by EPSRC grant EP/T015748/1.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

Overview

The talk will describe the Real Quantifier Elimination Problem, some applications in biology and economics, and some of the recent algorithmic developments that Coventry University has been involved in.

Joint work with: Chris Brown, Erika Abraham, Russell Bradford, Kelly Cohen, James Davenport, Tereso del Rio, Hassan Errami, Vladimir Gerdt, Dima Grigoriev, Charles Hoyt, Marek Kosta, Gereon Kremer, Casey Mulligan, Lynn Pickering, Jasper Nalbach, Ovidiu Radulescu, AmirHosein Sadeghi Manesh, Thomas Sturm, Zak Tonks and Andreas Weber.

Outline

Outline

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.

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 Tarski formulae example (Slide 3/58)

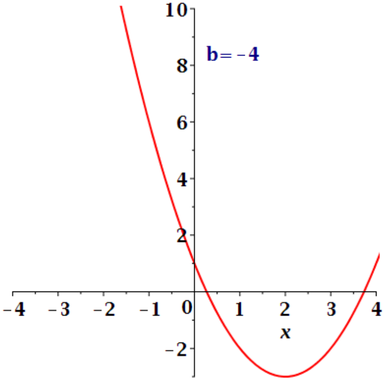
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example (Slide 3/58)

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

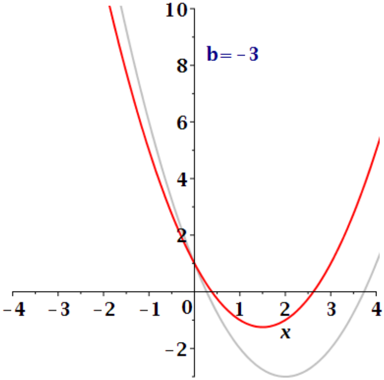
$$\forall x, x^2 + 1 > 0 \text{ is True },$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is False }.$$

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 automatically.



Partially quantified Tarski formulae example (Slide 3/58)

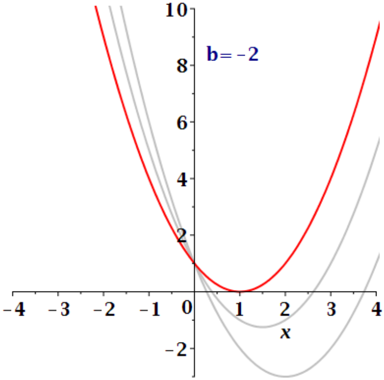
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example (Slide 3/58)

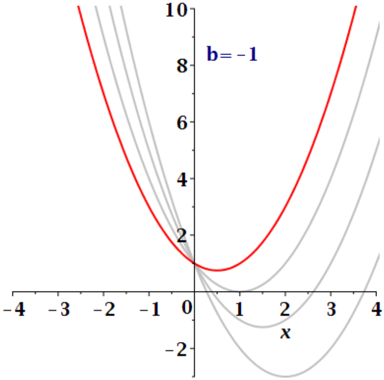
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example

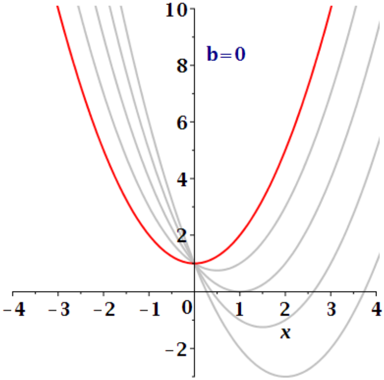
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example

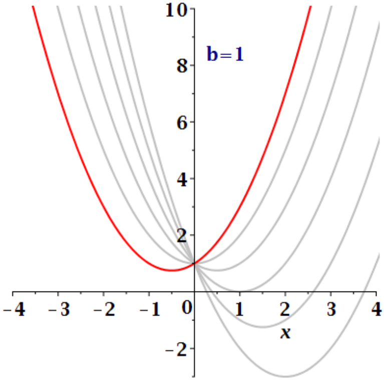
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example

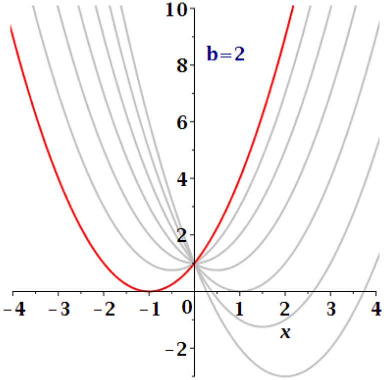
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example (Slide 3/58)

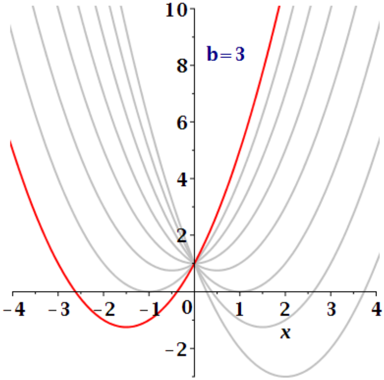
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Partially quantified Tarski formulae example

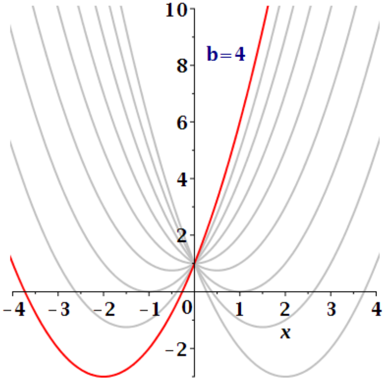
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

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

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 automatically.



Numeric vs. Symbolic solution?

The solution in the previous slide was essentially numeric: try what happens for many different values and interpolate a solution.

The alternative is to use **Symbolic Computation:** algorithms and data-structures for manipulating exact mathematical expressions and objects. Traditionally implemented in *Computer Algebra Systems* (e.g. Maple, Mathematica, Sage).

Advantages of Symbolic Computation:

- Can solve numerically ill-conditioned problems.
- Provide guarantees for safety critical systems.
- Provide fundamental insight into the system.

Disadvantage of Symbolic Computation: more expensive!

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

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 **cylindrical** meaning the projection of any two onto a lower coordinate space *in the variable ordering* are identical or disjoint (i.e. the cells in $\mathbb{R}^m$ stack up in cylinders over cells from CAD 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 finite set of sample points.

Most applications (e.g. QE) actually provide as input a logical formulae built from polynomial constraints and require as output a **truth-invariant CAD**: one such that each formula has constant truth value in each cell.
Such a CAD allows us to find solution sets from the descriptions of true cells: semi-algebraic; easy to visualise and check membership.

CAD Invariance Property

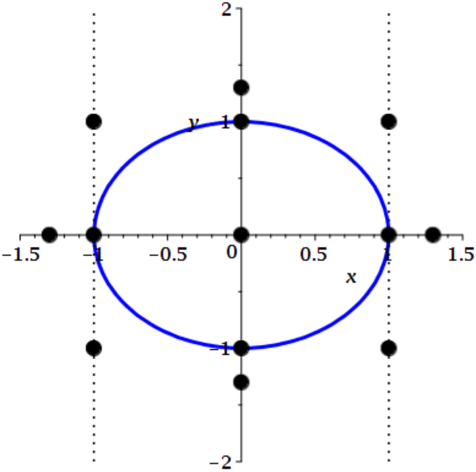
CAD may also refer to an algorithm that produces the CAD object.

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 finite set of sample points.

Most applications (e.g. QE) actually provide as input a logical formulae built from polynomial constraints and require as output a **truth-invariant CAD**: one such that each formula has constant truth value in each cell.
Such a CAD allows us to find solution sets from the descriptions of true cells: semi-algebraic; easy to visualise and check membership.

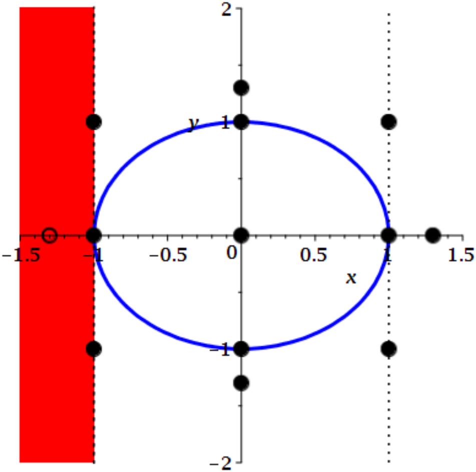
Example: Circle — visualisation

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



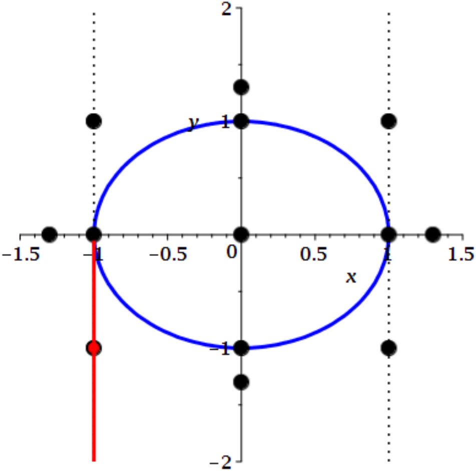
Example: Circle — visualisation

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



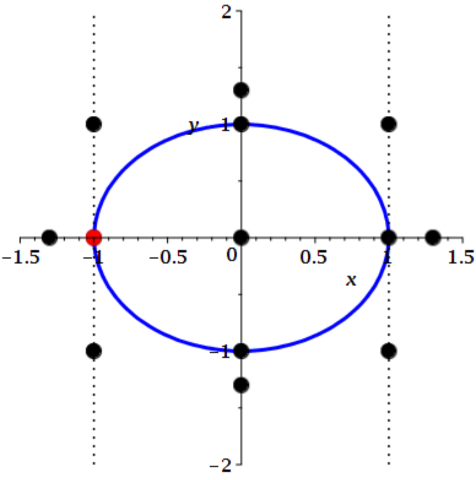
Example: Circle – visualisation

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



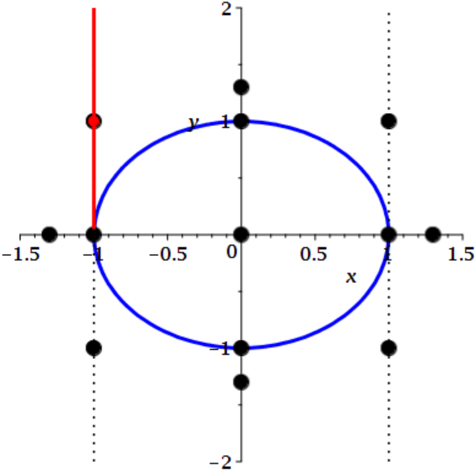
Example: Circle – visualisation

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



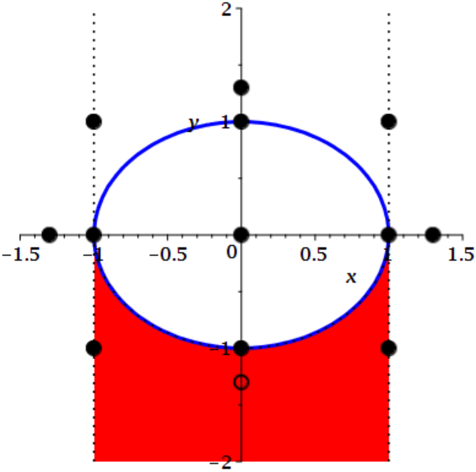
Example: Circle – visualisation

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



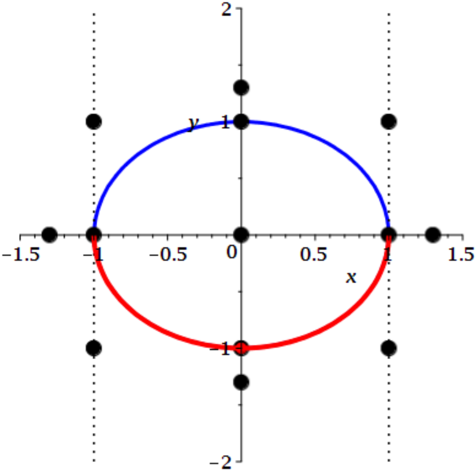
Example: Circle — visualisation

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



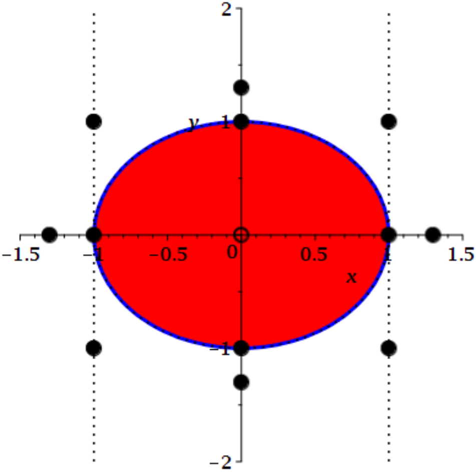
Example: Circle – visualisation

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

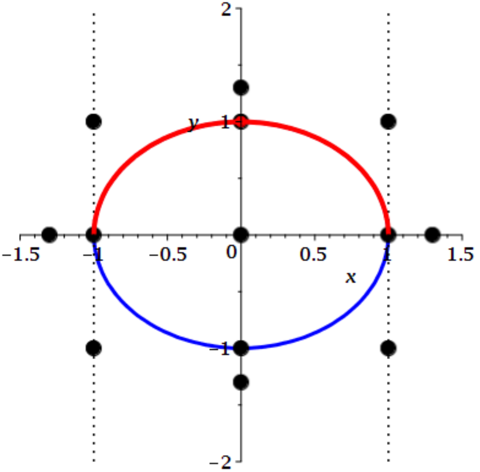
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x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

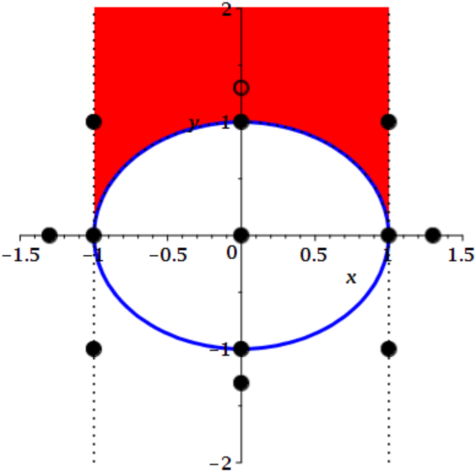
(Slide 7/58)

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



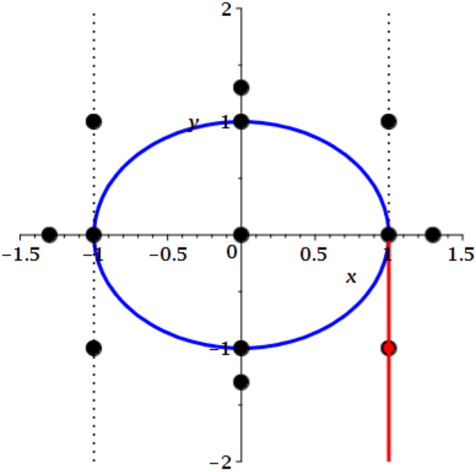
Example: Circle – visualisation

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



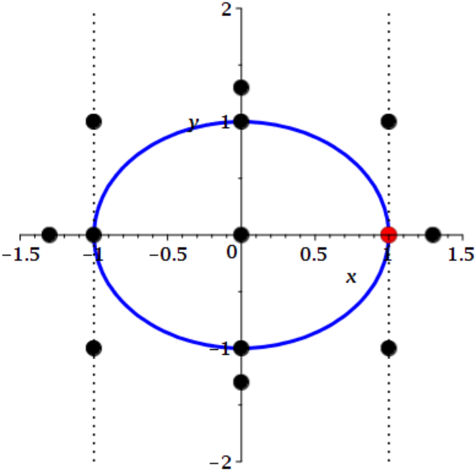
Example: Circle — visualisation

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



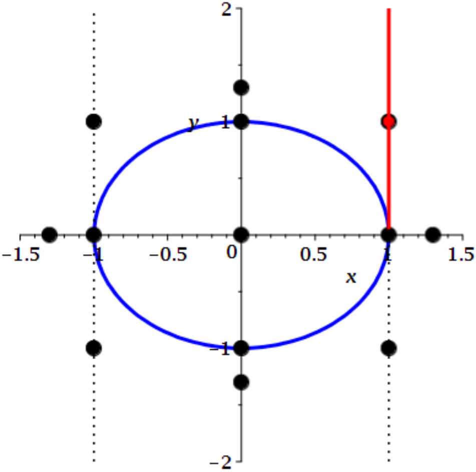
Example: Circle — visualisation (Slide 7/58)

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



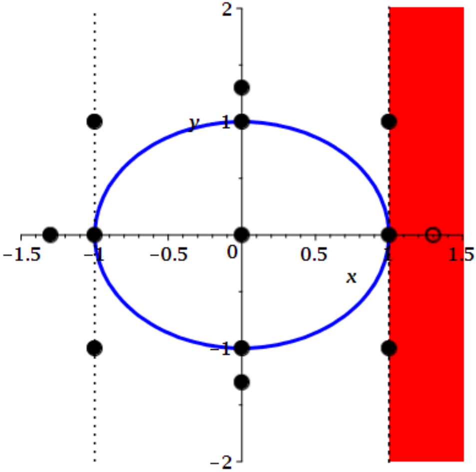
Example: Circle — visualisation

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



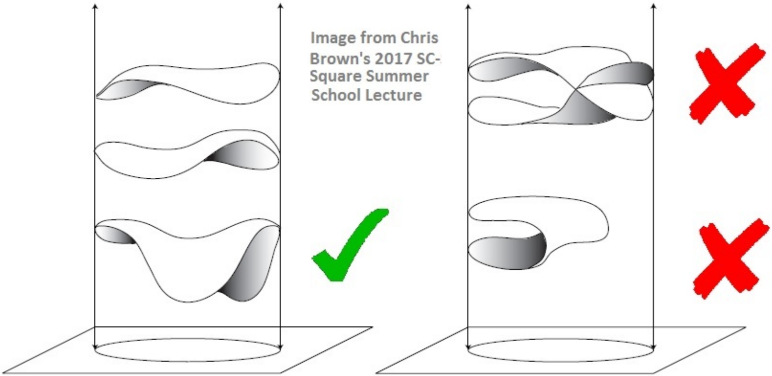
Example: Circle — visualisation

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?

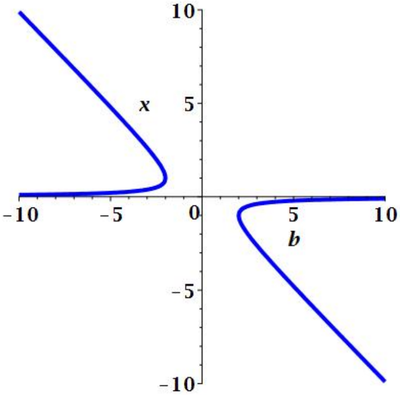
The usual approach is to calculate projection polynomials whose roots indicate changes in the behaviour of the input set; decompose with respect to these and then lift back working at a sample point. Safe if the projection provides **delineability**.



QE via CAD Example

How to determine with CAD?

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



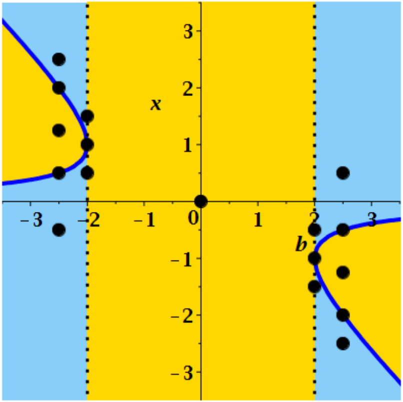
QE via CAD Example

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

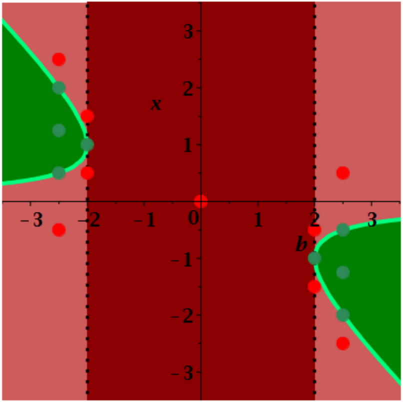
How to determine with CAD?

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

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?

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

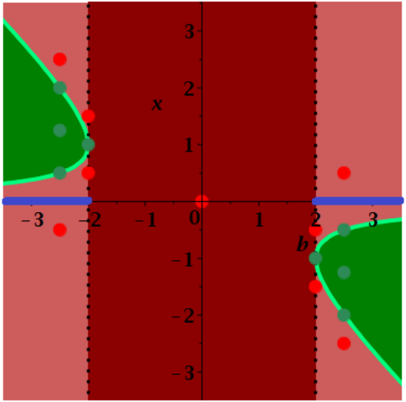
To solve we:

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:

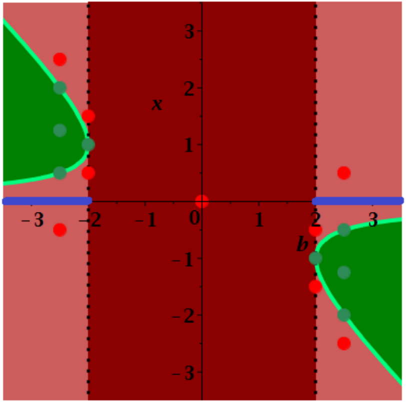
Build a sign-invariant CAD for $f = x^2 + bx + 1$.

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

Take disjunction of projections of true cells:

$$\implies$$

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



QE via CAD in General

In general we can perform Real QE on a decomposition as follows.

- Eliminate existential quantifiers by projecting the true cells, as in the previous example.
- Eliminate universal quantifiers by using the relation

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

  and then proceeding with existential QE.

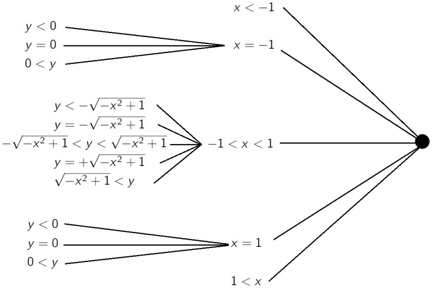
Recall our original example was $\forall x, x^2 + bx + 1 > 0$. The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$ which from the previous slide we know has solution $b \leq -2 \vee b \geq +2$. The solution to our universally quantified problem is then the negation of this: $-2 < b \wedge b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

Cylindricity Property

Decompositions allow us to understand infinite space with a finite number of samples and (semi-) algebraic cells allows us to construct solution formulae easily. But why cylindricity?

The Real QE solution process requires us to project cells (and combine those projections) and to calculate the complement of cells: both of these things are trivial for cells arranged cylindrically.

Cylindricity means we can think of CAD as a tree branching by variable.



CAD Complexity

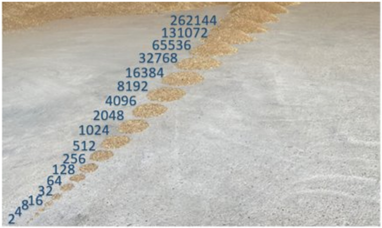
By the end of projection you have doubly exponentially many polynomials of doubly exponential degree (in the number of projections, i.e. variables). Hence also the number of real roots, cells and time to compute them grows doubly exponentially!

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

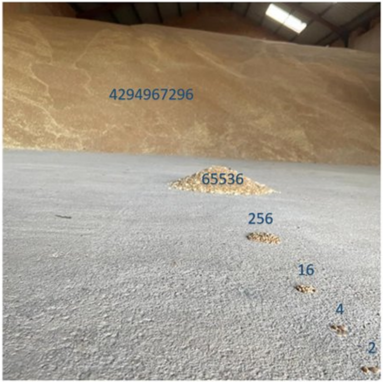
Thus in practice applications are only realistic for 4 or 5 variable, unless specific optimisations are utilised.

The Doubly Exponential Wall

Images by Tereso del Rio



Exponential Growth



Doubly Exponential Growth

QE via CAD Implementations

Both of the big proprietary Computer Algebra Systems have QE implementations: MATHEMATICA (the `Resolve` command) and MAPLE (inside `RegularChains:-SemiAlgebraicSetTools`).

The specialist computer algebra system QEPCADB 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 SAGE, and can now even run in your browser!

`http://tarski.tk/`

CAD implementations also exist in the SMT-solvers, SMT-RAT, YICES2, Z3 and CVC5. All are available for free – but note these can only answer fully quantified problems.

Alternatives to CAD for QE

There exists algorithms to achieve Real QE other than via CAD:

- Virtual Term Substitution: more efficient but degree limitations. Implementations in MATHEMATICA and REDLOG.
- QE via Comprehensive Gröbner Bases: more efficient if many equations. Implementation in SYNRAC package for MAPLE.
- Family of algorithms for Real QE with complexity doubly exponential in the number of quantifier eliminations. No implementations and analysis suggests that the crossover point where the asymptomatic growth becomes relevant is to far away for practical use.
- Variety of specialist Real QE algorithms for input of certain shape.

Alternatives to CAD for QE

There exists algorithms to achieve Real QE other than via CAD:

- Virtual Term Substitution: more efficient but degree limitations. Implementations in MATHEMATICA and REDLOG.
- QE via Comprehensive Gröbner Bases: more efficient if many equations. Implementation in SYNRAC package for MAPLE.
- Family of algorithms for Real QE with complexity doubly exponential in the number of quantifier eliminations. No implementations and analysis suggests that the crossover point where the asymptomatic growth becomes relevant is to far away for practical use.
- Variety of specialist Real QE algorithms for input of certain shape.

Alternatives to CAD for QE References

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, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 220–241. Springer-Verlag, 1998.

H.P. Le and M. Safey El Din.
Faster one block quantifier elimination for regular polynomial systems of equations.
In Proc. ISSAC '21, pages 265–272. ACM, 2021.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

Real QE Applications

QE can solve problems throughout engineering & science. E.g.

- derivation of optimal numerical schemes.
- artificial intelligence to pass university entrance exam.
- automated theorem proving.
- automated loop parellisation.
- structural design (minimising the weight of trusses).
- ⋮

There is no lack of potential applications: the problem is scaling up in the face of high complexity!

QE Applications References

- M. Erascu and H. Hong.
  Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation).
  Journal of Symbolic Computation, 75:110–126, 2016.
- N.H. Arai, T. Matsuzaki, H. Iwane, and H. Anai.
  Mathematics by machine.
  In Proc. ISSAC '14, pages 1–8. ACM, 2014.
- L.C. Paulson.
  Metitarski: Past and future.
  In Proc ITP '12, pages 1–10. Springer, 2012.
- A. Grosslinger, M. Griebl, and C. Lengauer.
  Quantifier elimination in automatic loop parallelization.
  Journal of Symbolic Computation, 41(11):1206–1221, 2006.
- A.E. Charalampakis and I. Chatzigiannelis.
  Analytical solutions for the minimum weight design of trusses by cylindrical algebraic decomposition.
  Archive of Applied Mechanics, 88(1):39–49, 2018.

Outline

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \; . \qquad (3X_2 \xrightarrow{k_4} 2X_1 + X_2)$$

Species: X_1, X_2.
Reactions:

$$\begin{aligned} 2X_1 + X_2 &\xrightarrow{k_1} 3X_1 \\ 3X_1 &\xrightarrow{k_2} X_1 + 2X_2 \\ X_1 + 2X_2 &\xrightarrow{k_3} 3X_2 \\ 3X_2 &\xrightarrow{k_4} 2X_1 + X_2 \end{aligned}$$

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \; . \quad (3X_2 \xrightarrow{k_4} 2X_1 + X_2)$$

From this we define a dynamical system:

$$\begin{cases} \dot{x_1} = (3-2)k_1 x_1 x_1 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3, \\ \dot{x_2} = -k_1 x_1^2 x_2 + 2k_2 x_1^3 + k_3 x_1 x_2^2 - 2k_4 x_2^3. \end{cases}$$

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \; .$$
$$3X_2 \xrightarrow{k_4} 2X_1 + X_2$$

From this we define a dynamical system:

$$\begin{cases} \dot{x_1} = k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3, \\ \dot{x_2} = -k_1 x_1^2 x_2 + 2k_2 x_1^3 + k_3 x_1 x_2^2 - 2k_4 x_2^3. \end{cases}$$

Note that the sum of the derivatives is zero. We thus have a *conservation law*, that the sum of x_1 and x_2 is a constant:

$$x_1 + x_2 = k_5.$$

CRN Steady States

A CRN has steady states when all the derivatives are zero. For our toy example these occur where

$$\begin{cases} k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3 = 0, \\ x_1 + x_2 - k_5 = 0. \end{cases}$$

In this system we have

- Variables, the x_is: concentrations of species.
- Parameters, the k_is: reaction rate constants and constants of conservation laws.

All of these must be non-negative real numbers. Although not quantified, we may still use CAD to identify and describe the solution regions.

Multistationarity

A CRN exhibits **multistationarity** if there exists a choice of parameter values for which the system has more than one (positive real) solution.

Why care about multistationarity?

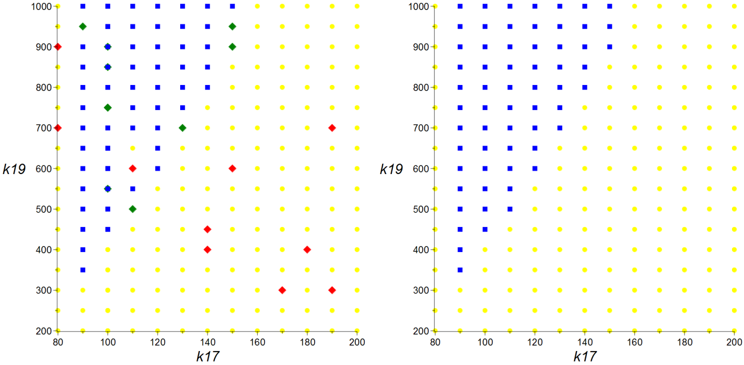
- Used for switch behaviour.
- Instrumental to cellular memory and cell differentiation during development.
- Used by micro organisms in survival strategies.
- Used in decision making processes in the cell division cycle.

There are a variety of efficient methods for answering the Boolean question as to whether or not a system can exhibit multistationarity. Less studied is the question of determining the actual parameter values where multistationarity occurs.

Numeric Methods for Multistationarity 1

Numeric sampling of the parameter space is used. However:

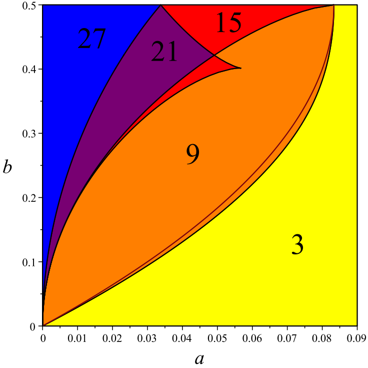
- Incorrect results can be obtained at ill-conditioned points.
- No guarantee all areas of interest will be sampled.



Numeric Methods for Multistationarity 2

Numeric sampling of the parameter space is used. However:

- Incorrect results can be obtained at ill-conditioned points.
- No guarantee all areas of interest will be sampled.



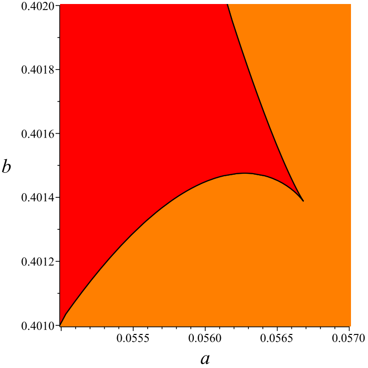


Image from AmirHosein SadeghiManesh.

Symbolic Methods for Multistationarity

The identification of regions of multistationarity can be achieved by symbolic methods for non-linear real arithmetic.

For our toy example, symbolic tools can determine that the semi-algebraic system

$$\begin{cases} k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3 = 0, \\ x_1 + x_2 - k_5 = 0, \\ x_1, x_2, k_1, k_2, k_3, k_4, k_5 > 0, \end{cases}$$

has multiple solutions if and only if

$$8k_1^3 k_4 - k_1^2 k_3^2 - 72 k_1 k_2 k_3 k_4 + 432 k_2^2 k_4^2 + 8 k_2 k_3^2 < 0.$$

What about a larger example?

Case Study: Model 26

From: `www.ebi.ac.uk/biomodels-main/BIOMD0000000026`

$$\dot{x_1} = k_2x_6 + k_{15}x_{11} - k_1x_1x_4 - k_{16}x_1x_5$$
$$\dot{x_2} = k_3x_6 + k_5x_7 + k_{10}x_9 + k_{13}x_{10} - x_2x_5(k_{11} + k_{12}) - k_4x_2x_4$$
$$\dot{x_3} = k_6x_7 + k_8x_8 - k_7x_3x_5$$
$$\dot{x_4} = x_6(k_2 + k_3) + x_7(k_5 + k_6) - k_1x_1x_4 - k_4x_2x_4$$
$$\dot{x_5} = k_8x_8 + k_{10}x_9 + k_{13}x_{10} + k_{15}x_{11} - x_2x_5(k_{11} + k_{12}) - k_7x_3x_5 - k_{16}x_1x_5$$
$$\dot{x_6} = k_1x_1x_4 - x_6(k_2 + k_3)$$
$$\dot{x_7} = k_4x_2x_4 - x_7(k_5 + k_6)$$
$$\dot{x_8} = k_7x_3x_5 - x_8(k_8 + k_9)$$
$$\dot{x_9} = k_9x_8 - k_{10}x_9 + k_{11}x_2x_5$$
$$\dot{x_{10}} = k_{12}x_2x_5 - x_{10}(k_{13} + k_{14})$$
$$\dot{x_{11}} = k_{14}x_{10} - k_{15}x_{11} + k_{16}x_1x_5$$

11 differential equations

11 variables

16 parameters

Parameters and Conservation Laws

The Biomodels database gives reliable estimates of the rate constants for this example.

$$k_1 = 0.02, \quad k_3 = 0.01, \quad k_4 = 0.032,$$
$$k_7 = 0.045, \quad k_9 = 0.092, \quad k_{11} = 0.01,$$
$$k_{12} = 0.01, \quad k_{15} = 0.086, \quad k_{16} = 0.0011.$$

$$k_2 = 1, \quad k_5 = 1, \quad k_6 = 15, \quad k_8 = 1,$$
$$k_{10} = 1, \quad k_{13} = 1, \quad k_{14} = 0.5.$$

Three Linear Conservation Constraints may be derived:

$$x_5 + x_8 + x_9 + x_{10} + x_{11} = k_{17}$$
$$x_4 + x_6 + x_7 = k_{18}$$
$$x_1 + x_2 + x_3 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} = k_{19}$$

These introduce new parameters for which there are not good estimates. Treat as symbolic parameters.

Algebraic System of Interest

$$
\begin{aligned}
0 &= -200x_1x_4 - 11x_1x_5 + 860x_{11} + 10000x_6,\\
0 &= -16x_2x_4 - 10x_2x_5 + 500x_{10} + 5x_6 + 500x_7 + 500x_9,\\
0 &= -9x_3x_5 + 3000x_7 + 200x_8,\\
0 &= -10x_1x_4 - 16x_2x_4 + 505x_6 + 8000x_7,\\
0 &= -11x_1x_5 - 200x_2x_5 - 450x_3x_5 + 10000x_{10} + 860x_{11} + 10000x_8 + 10000x_9,\\
0 &= 2x_1x_4 - 101x_6,\\
0 &= 4x_2x_4 - 2000x_7,\\
0 &= 45x_3x_5 - 1092x_8,\\
0 &= 5x_2x_5 + 46x_8 - 500x_9,\\
0 &= x_2x_5 - 150x_{10},\\
0 &= 11x_1x_5 + 5000x_{10} - 860x_{11},\\
0 &= -k_{17} + x_{10} + x_{11} + x_5 + x_8 + x_9,\\
0 &= -k_{18} + x_4 + x_6 + x_7,\\
0 &= -k_{19} + x_1 + x_{10} + x_{11} + x_2 + x_3 + x_6 + x_7 + x_8 + x_9,\\
0 &< x_1,\ \ldots,\ 0 < x_{11}, 0 < k_{17}, 0 < k_{18}, 0 < k_{19}.
\end{aligned}
$$

14 polynomial equations
14 positivity conditions
11 variables
3 parameters

Algebraic System of Interest

$$
\begin{aligned}
0 &= -200x_1x_4 - 11x_1x_5 + 860x_{11} + 10000x_6,\\
0 &= -16x_2x_4 - 10x_2x_5 + 500x_{10} + 5x_6 + 500x_7 + 500x_9,\\
0 &= -9x_3x_5 + 3000x_7 + 200x_8,\\
0 &= -10x_1x_4 - 16x_2x_4 + 505x_6 + 8000x_7,\\
0 &= -11x_1x_5 - 200x_2x_5 - 450x_3x_5 + 10000x_{10} + 860x_{11} + 10000x_8 + 10000x_9,\\
0 &= 2x_1x_4 - 101x_6,\\
0 &= 4x_2x_4 - 2000x_7,\\
0 &= 45x_3x_5 - 1092x_8,\\
0 &= 5x_2x_5 + 46x_8 - 500x_9,\\
0 &= x_2x_5 - 150x_{10},\\
0 &= 11x_1x_5 + 5000x_{10} - 860x_{11},\\
0 &= -k_{17} + x_{10} + x_{11} + x_5 + x_8 + x_9,\\
0 &= -k_{18} + x_4 + x_6 + x_7,\\
0 &= -k_{19} + x_1 + x_{10} + x_{11} + x_2 + x_3 + x_6 + x_7 + x_8 + x_9,\\
0 &< x_1, \ldots, 0 < x_{11}, 0 < k_{17}, 0 < k_{18}, 0 < k_{19}.
\end{aligned}
$$

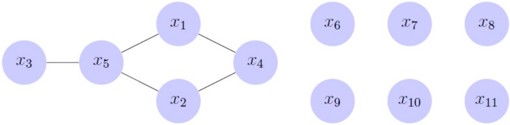
14 polynomial equations
14 positivity conditions
11 variables
3 parameters
14 symbolic indeterminates

Symbolic Methods I of IV

We seek a decomposition of the parameter space so CAD is a potential tool. But this example is past the doubly exponential wall. We need something else first.

1. CRN models often have low total degrees with many linear monomials. Idea: pre-processing input with **parametric Gaussian Elimination** (pGE): i.e. solving suitable equations with respect to some variable and substituting the corresponding solution into the system.
   Requires case distinctions but these often cancelled out by the positivity constraints.

Model 26: After pGE Pre-processing (Slide 27/58)



$$
\begin{aligned}
\overline{\psi} = x_5 > 0 \wedge x_4 > 0 \wedge k_{19} > 0 \wedge k_{18} > 0 \wedge k_{17} > 0 \\
\wedge\, & 1062444 k_{18} x_4^2 x_5 + 23478000 k_{18} x_4^2 + 1153450 k_{18} x_4 x_5^2 + 2967000 k_{18} x_4 x_5 \\
& + 638825 k_{18} x_5^3 + 49944500 k_{18} x_5^2 - 5934 k_{19} x_4^2 x_5 - 989000 k_{19} x_4 x_5^2 \\
& - 1062444 x_4^3 x_5 - 23478000 x_4^3 - 1153450 x_4^2 x_5^2 - 2967000 x_4^2 x_5 \\
& - 638825 x_4 x_5^3 - 49944500 x_4 x_5^2 = 0 \\
\wedge\, & 1062444 k_{17} x_4^2 x_5 + 23478000 k_{17} x_4^2 + 1153450 k_{17} x_4 x_5^2 + 2967000 k_{17} x_4 x_5 \\
& + 638825 k_{17} x_5^3 + 49944500 k_{17} x_5^2 - 1056510 k_{19} x_4^2 x_5 - 164450 k_{19} x_4 x_5^2 \\
& - 638825 k_{19} x_5^3 - 1062444 x_4^2 x_5^2 - 23478000 x_4^2 x_5 - 1153450 x_4 x_5^3 \\
& - 2967000 x_4 x_5^2 - 638825 x_5^4 - 49944500 x_5^3 = 0.
\end{aligned}
$$

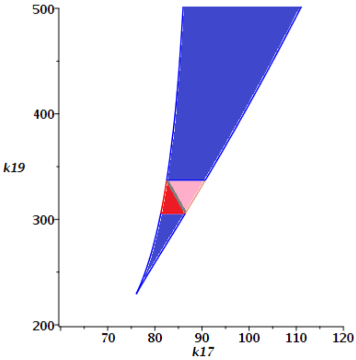
Symbolic Methods II of IV

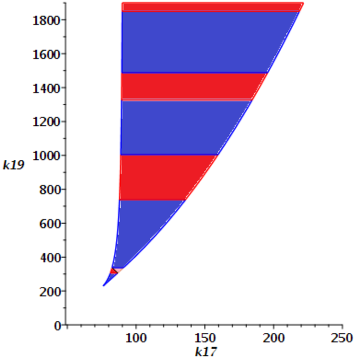
C.Chen, J.Davenport, J.May, M.Moreno Maza, B.Xia, R.Xiao.
Triangular decomposition of semi-algebraic systems.
Journal of Symbolic Computation, 49:3–26, 2013.

2. A Real Triangularization is a decomposition of a polynomial system into finitely many regular semi-algebraic systems. Real counterparts of the well studied triangular decompositions over $\mathbb{C}$.
   The paper that introduced them also outlined a Lazy variant of the algorithm which produces the highest dimension component and unevaluated function calls for the rest (guaranteed of lower dimension). If we just use the former we are blind to what happens on the boundaries, but this is not usually relevant to biology applications.

Model 26: LRT and Open CAD

With two free parameters we can perform a Lazy Real Triangularize and then an (open) CAD of the largest component.





Reference

Approaches 1 and 2 appear here:

R. Bradford, J.H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Košta, O. Radulescu, T. Sturm, and A. Weber.
Identifying the parametric occurrence of multiple steady states for some biological networks.
Journal of Symbolic Computation, 98:84–119, 2020.

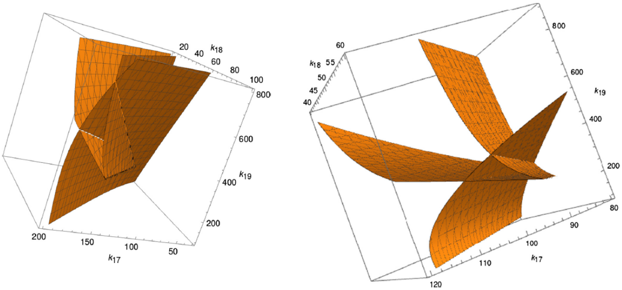
Symbolic Methods III of IV

D. Lazard and F. Rouillier.
Solving parametric polynomial systems.
Journal of Symbolic Computation, 42(6):636–667, 2007.

3. The **Discriminant Variety** was introduced in 2009 as the variety in a parameter space over which a system of equations has invariant number of solutions. Thus a CAD of such a variety would provide the decomposition we need. Computed via a Gröbner Basis calculation on the polynomials and the determinant of the Jacobian matrix.

Model 26: Discriminant Variety

D. Lichtblau.
Symbolic analysis of multiple steady states in a MAPK chemical reaction network.
Journal of Symbolic Computation, 105:118–144, 2021.



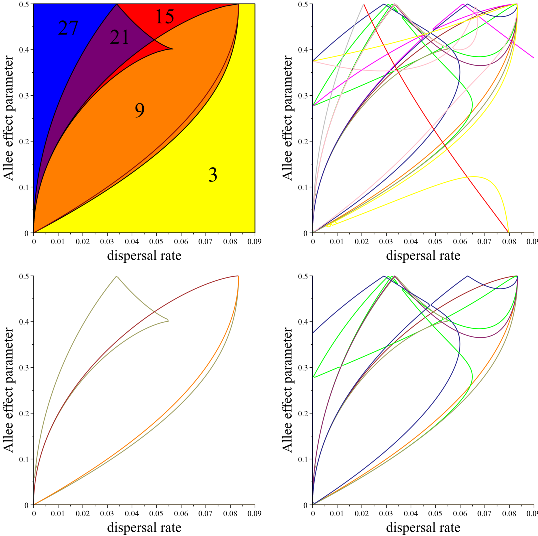
Symbolic Methods IV of IV

4. The **Discriminant Variety** is the optimal object to perform the decomposition upon. However, it can be expensive. A superset containing it can be produced through resultant methods.

A. Sadeghimanesh and M. England.
Resultant Tools for Parametric Polynomial Systems with Application to Population Models.
Submitted. Preprint at arXiv:2201.13189.

Paper applies the approach to a problem from population dynamics for which calculating the discriminant variety with GB was infeasible. But the superset could be constructed and decomposed easily to allow for analysis by CAD.

Discriminant Variety via Resultants Example



Symbolic-Numeric Approaches Possible

Scope for symbolic-numeric methods:

M. England, H. Errami, D. Grigoriev, O. Radulescu, T. Sturm, and A. Weber.
Symbolic versus numerical computation and visualization of parameter regions for multistationarity of biological networks.
In Proc. CASC '17, pages 93–108. Springer, 2017.

G. Röst and A. Sadeghimanesh.
Exotic bifurcations in three connected populations with Allee effects.
International Journal of Bifurcation and Chaos, 31(13):2150202, 2021.

Scope for numeric sampling based decomposition approach.

E. Feliu and A. Sadeghimanesh.
Kac-Rice formulas and the number of soluions of parametrized systems of polynomial equations.
Mathematics of Computation, 91:338, 2022.

A. Sadeghimanesh and M. England.
Polynomial Superlevel Set Representation of the Multistationarity Region of Chemical Reaction Networks.
BMC Bioinformatics, 23 Article number 391, 26 pages, 2022.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

Framework for Reasoning in Economics

Determine whether, with variables $\Lambda = (v_1, \ldots, v_n)$, the hypotheses $H(\Lambda)$ follow from the assumptions $A(\Lambda)$. I.e. answer

$$\forall \Lambda \,.\, A(\Lambda) \Rightarrow H(\Lambda)?$$

Logically the answer must be `True` or `False`. But economists are interested also in the following:

- Are the assumptions themselves contradictory?
- If `False`, can additional assumptions be made to give `True`?
- If `True`, can any assumptions be removed?

Such questions can be answered by Real QE in many cases.

Categorisation by a pair of statements

THEORYGURU package for MATHEMATICA led by Casey Milligan (Chicago). For a proposed economics theorem, we check both:

- the existence of an example
  $\exists \Lambda, A \wedge H$,
- and the existence of a counterexample
  $\exists \Lambda, A \wedge \neg H$.

Then we can categorize the proposed theorem as follows:

	$\neg\exists \Lambda, A \wedge \neg H$	$\exists \Lambda, A \wedge \neg H$
$\exists \Lambda, A \wedge H$	True	Mixed
$\neg\exists \Lambda, A \wedge H$	Contradictory Assumptions	False

Theory Exploration

An economist could vary the question by strengthening the assumptions that led to a `Mixed` result in search of a `True` theorem; or weaken the assumptions that generated a `True` result to identify a theorem that can be applied more widely.

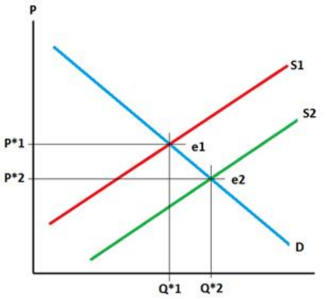
This may be discovered by quantifying more or less of the variables in Λ. For example, partition Λ as Λ_1, Λ_2 and ask for

$$\{\Lambda_1 : \forall \Lambda_2 \,.\, A(\Lambda_1, \Lambda_2) \Rightarrow H(\Lambda_1, \Lambda_2)\}.$$

This can be tackled with Real QE to result in a formula in the free variables that can be used to modify the assumptions accordingly.

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, a, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price*.

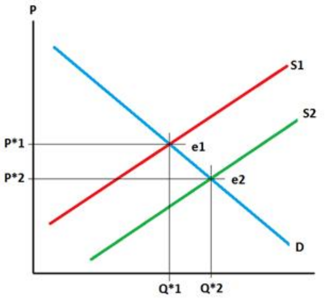


$$A \equiv D'(q) < 0 \wedge S'(q) > 0 \wedge \frac{dp}{da} = \frac{d}{da}(S(q) - a) \wedge \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \wedge \frac{dp}{da} < 0$$

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, *a*, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price*.

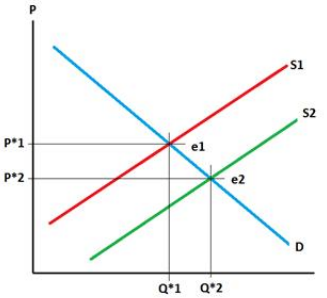


$$A \equiv D'(q) < 0 \wedge S'(q) > 0 \wedge \frac{dp}{da} = \frac{d}{da}(S(q) - a) \wedge \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \wedge \frac{dp}{da} < 0$$

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, a, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price.*

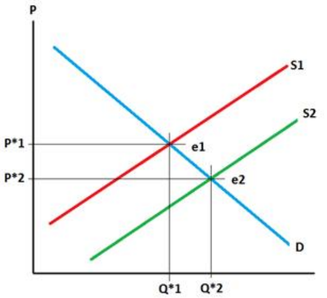


$$A \equiv D'(q) < 0 \wedge S'(q) > 0 \wedge \frac{dp}{da} = \frac{d}{da}(S(q) - a) \wedge \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \wedge \frac{dp}{da} < 0$$

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, a, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price*.



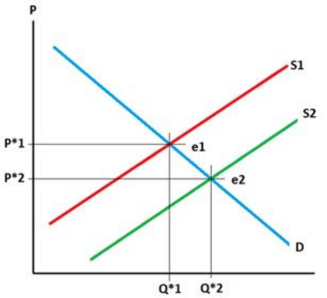
$$A \equiv D'(q) < 0 \land S'(q) > 0 \land \frac{dp}{da} = \frac{d}{da}(S(q) - a) \land \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \land \frac{dp}{da} < 0$$

Here $\frac{dq}{da}$ is the quantity impact and $\frac{dp}{da}$ the price impact of the cost reduction – describing the move in equilibrium.

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, a, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price*.



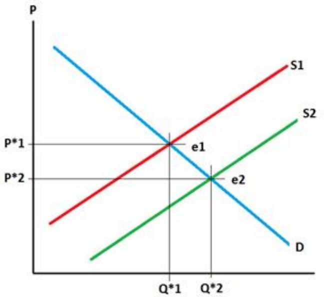
$$A \equiv D'(q) < 0 \wedge S'(q) > 0 \wedge \frac{dp}{da} = \frac{d}{da}(S(q) - a) \wedge \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \wedge \frac{dp}{da} < 0$$

Here $\frac{dq}{da}$ is the quantity impact and $\frac{dp}{da}$ the price impact of the cost reduction – describing the move in equilibrium.

Simple Example: Marshall 1895

Marshall considered the effect of a reduction in supply costs, a, concluding that *for any supply-demand equilibrium in which the two curves have their usual slopes, a downward supply shift increases the equilibrium quantity and decreases the equilibrium price*.



$$A \equiv D'(q) < 0 \wedge S'(q) > 0 \wedge \frac{dp}{da} = \frac{d}{da}(S(q) - a) \wedge \frac{dp}{da} = \frac{d}{da}D(q)$$

$$H \equiv \frac{dq}{da} > 0 \wedge \frac{dp}{da} < 0$$

Here $\frac{dq}{da}$ is the quantity impact and $\frac{dp}{da}$ the price impact of the cost reduction – describing the move in equilibrium.

Marshall as Real QE

To study as a Tarski formula we set the "variables" Λ to be four real numbers (v_1, v_2, v_3, v_4):

$$\Lambda = \left\{ D'(q), S'(q), \frac{dq}{da}, \frac{dp}{da} \right\}.$$

Then, after applying the chain rule, A and H may be understood as Boolean combinations of polynomial equalities and inequalities:

$$A \equiv v_1 < 0 \land v_2 > 0 \land v_4 = v_3 v_2 - 1 \land v_4 = v_3 v_1,$$
$$H \equiv v_3 > 0 \land v_4 < 0.$$

From here it takes only a little reasoning by hand to see that $A \Rightarrow H$. Any of the Real QE tools mentioned earlier can conclude this almost instantly.

Marshall as Real QE

To study as a Tarski formula we set the "variables" Λ to be four real numbers (v_1, v_2, v_3, v_4):

$$\Lambda = \left\{ D'(q), S'(q), \frac{dq}{da}, \frac{dp}{da} \right\}.$$

Then, after applying the chain rule, A and H may be understood as Boolean combinations of polynomial equalities and inequalities:

$$A \equiv v_1 < 0 \wedge v_2 > 0 \wedge v_4 = v_3 v_2 - 1 \wedge v_4 = v_3 v_1,$$
$$H \equiv v_3 > 0 \wedge v_4 < 0.$$

From here it takes only a little reasoning by hand to see that $A \Rightarrow H$. Any of the Real QE tools mentioned earlier can conclude this almost instantly.

Marshall as Real QE

To study as a Tarski formula we set the "variables" Λ to be four real numbers (v_1, v_2, v_3, v_4):

$$\Lambda = \left\{ D'(q), S'(q), \frac{dq}{da}, \frac{dp}{da} \right\}.$$

Then, after applying the chain rule, A and H may be understood as Boolean combinations of polynomial equalities and inequalities:

$$A \equiv v_1 < 0 \land v_2 > 0 \land v_4 = v_3 v_2 - 1 \land v_4 = v_3 v_1,$$
$$H \equiv v_3 > 0 \land v_4 < 0.$$

From here it takes only a little reasoning by hand to see that $A \Rightarrow H$. Any of the Real QE tools mentioned earlier can conclude this almost instantly.

Example #0013 Context

Context: Disagreement between Mulligan and Krugman on causes of recession. In particular, latter asserted that whenever taxes on labour supply are primarily responsible for a recession then wages increase. Benchmark Example #0013 considers this claim.

Two scenarios to track: what actually happens (act) when taxes (t) and demand forces (a) together create a recession, and what would have happened (hyp) if taxes on labour supply had been the only factor affecting the labour market.

The labour demand and supply functions $D(w, a)$ and $S(w, t)$ meet at the labour market equilibrium to supply quantity of labour q.

Example #0013 Context

Context: Disagreement between Mulligan and Krugman on causes of recession. In particular, latter asserted that whenever taxes on labour supply are primarily responsible for a recession then wages increase. Benchmark Example #0013 considers this claim.

Two scenarios to track: what actually happens (*act*) when taxes (t) and demand forces (a) together create a recession, and what would have happened (*hyp*) if taxes on labour supply had been the only factor affecting the labour market.

The labour demand and supply functions $D(w, a)$ and $S(w, t)$ meet at the labour market equilibrium to supply quantity of labour q.

Example #0013 Logic

$$A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0$$
$$\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1$$
$$\wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right)$$
$$\wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right)$$
$$\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0$$
$$\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0$$
$$H \equiv \frac{dw}{dact} > 0.$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\big(D(w,a) = q = S(w,t)\big) \\
& \wedge \frac{d}{dhyp}\big(D(w,a) = q = S(w,t)\big) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0$$
$$\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1$$
$$\wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right)$$
$$\wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right)$$
$$\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0$$
$$\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0$$
$$H \equiv \frac{dw}{dact} > 0.$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0$$
$$\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1$$
$$\wedge \frac{d}{dact}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0$$
$$\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0$$
$$H \equiv \frac{dw}{dact} > 0.$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0
\end{aligned}
$$

$$
H \equiv \frac{dw}{dact} > 0.
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Result

We may view this as a Tarski formula in 12 variables,

$$\Lambda = \left\{ \frac{da}{dact}, \frac{da}{dhyp}, \frac{dt}{dact}, \frac{dt}{dhyp}, \frac{dq}{dact}, \frac{dq}{dhyp}, \frac{dw}{dact}, \frac{dw}{dhyp}, \frac{\partial D(w,a)}{\partial a}, \frac{\partial S(w,t)}{\partial t}, \frac{\partial D(w,a)}{\partial w}, \frac{\partial S(w,t)}{\partial w} \right\},$$

Each is representing a partial derivative describing the supply and demand function or a total derivative indicating a change over time within a scenario.

Evaluating the two existential problems shows that both examples and counterexamples exist: the theorem is not universally true.

Example #0013 Exploration

If we leave $\frac{\partial D(w,a)}{\partial w}$ and $\frac{\partial S(w,t)}{\partial w}$ as free variables then QE recovers a disjunction of three quantifier-free formulae. Two of them contradict the assumptions, but the third, below, could be added to A to guarantee the truth of H.

$$\frac{\partial S(w,t)}{\partial w} \geq -\frac{\partial D(w,a)}{\partial w} > 0.$$

The added assumption states that labour supply is at least as sensitive to wages as labour demand.

Economics Example References

C. Mulligan, R. Bradford, J.H. Davenport, M. England, and Z. Tonks.
Non-linear real arithmetic benchmarks derived from automated reasoning in economics.
In Proc. SC^2 '1), CEUR-WS 2189, pages 48–60, 2018.
URL: http://ceur-ws.org/Vol-2189/.

C.B. Mulligan, J.H. Davenport, and M. England.
TheoryGuru: A Mathematica package to apply quantifier elimination technology to economics.
In Proc. ICMS 2018, LNCS 10931, pages 369–378. Springer, 2018.
URL: https://doi.org/10.1007/978-3-319-96418-8_44.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
2. Some Recent Applications
   - Biology
   - Economics
3. New Algorithmic Developments
   - Redesigning CAD for SMT
   - Machine Learning for Real QE

Satisfiability in NRA

We now consider the problem of determining the satisfiability of a formula in Non-linear Real Arithmetic (NRA). I.e. to evaluate

$$\exists x_1, \exists x_2, \ldots, \exists x_n F(x_1, x_2, \ldots, x_n)$$

as either True (SAT) or False (UNSAT) where F is a formula in Boolean logic (atoms connected by $\wedge, \vee, \neg$) whose atoms are sign constraints on non-linear multivariate polynomials with integer coefficients. (Usually assume F is in conjunctive normal form.)

This is a sub-problem of Real QE and thus can be solved by CAD etc. But this problem has lower (single exponential) complexity.

A sign-invariant CAD for the polynomials in F can be used to solve any such problem, regardless of the particular Boolean structure involved. How to adapt CAD to take note of the logic?

Satisfiability in NRA

We now consider the problem of determining the satisfiability of a formula in Non-linear Real Arithmetic (NRA). I.e. to evaluate

$$\exists x_1, \exists x_2, \ldots, \exists x_n F(x_1, x_2, \ldots, x_n)$$

as either True (SAT) or False (UNSAT) where F is a formula in Boolean logic (atoms connected by $\wedge, \vee, \neg$) whose atoms are sign constraints on non-linear multivariate polynomials with integer coefficients. (Usually assume F is in conjunctive normal form.)

This is a sub-problem of Real QE and thus can be solved by CAD etc. But this problem has lower (single exponential) complexity.

A sign-invariant CAD for the polynomials in F can be used to solve any such problem, regardless of the particular Boolean structure involved. How to adapt CAD to take note of the logic?

The SMT Approach

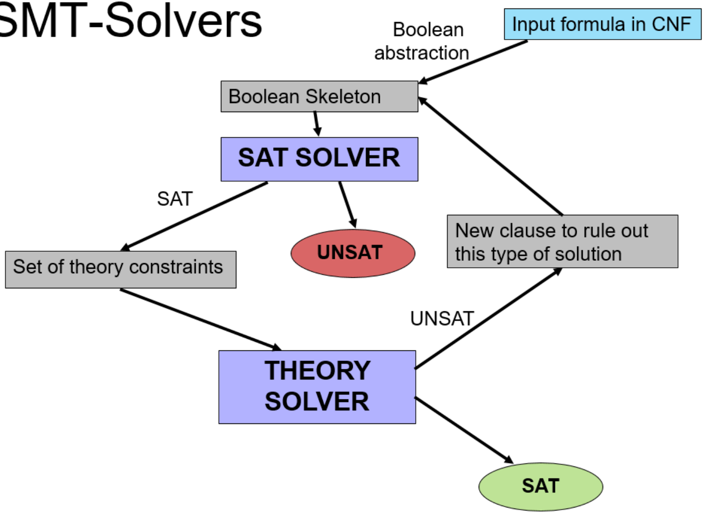
One approach to such satisfiability problems is to separate out the logic from the arithmetic theory.

- Allow the logical structure to be explored by a **SAT Solver**.
- Have the solutions proposed be tested in the theory of interest by relevant software for that domain: a **Theory Solver**.

The Theory Solver need only test the consistency of a set of constraints (no Boolean logic).

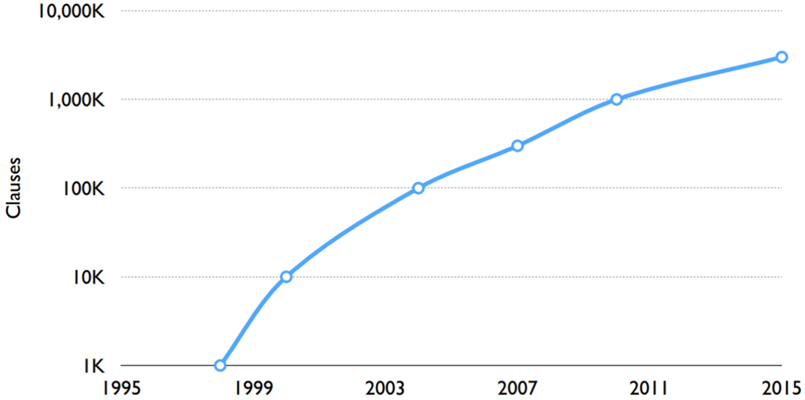
This approach is called **Satisfiability Modulo Theories** (SMT).

SMT-Solvers


Input formula in CNF
Boolean abstraction
Boolean Skeleton
SAT SOLVER
SAT
UNSAT
Set of theory constraints
New clause to rule out this type of solution
UNSAT
THEORY SOLVER
SAT

Why the SMT Approach?

To take advantage of the incredible progress in SAT solvers!



Based on a slide from Vijay Ganesh

CAD as NRA Theory Solver

We can use CAD as SMT NRA theory solver but it must be adapted for this use:

- **Incrementality:** Add a constraint and divide cells.
- **Backtracking:** Remove a constraint and merge cells.
- **Explanations:** When no cell satisfies constraints identify minimal subset of constraints which are mutually unsatisfiable.

G. Kremer and E. Ábrahám.
Fully incremental cylindrical algebraic decomposition.
J. of Symbolic Computation, 100, pages 11–37. Elsevier, 2020.
https://doi.org/10.1016/j.jsc.2019.07.018

(Q) Why is SAT solver + CAD better than CAD alone?
(A) Because in SMT a theory solver commonly only addresses a small subset of the total constraints.

CAD as NRA Theory Solver

We can use CAD as SMT NRA theory solver but it must be adapted for this use:

- **Incrementality:** Add a constraint and divide cells.
- **Backtracking:** Remove a constraint and merge cells.
- **Explanations:** When no cell satisfies constraints identify minimal subset of constraints which are mutually unsatisfiable.

G. Kremer and E. Ábrahám.
Fully incremental cylindrical algebraic decomposition.
J. of Symbolic Computation, 100, pages 11–37. Elsevier, 2020.
https://doi.org/10.1016/j.jsc.2019.07.018

(Q) Why is SAT solver + CAD better than CAD alone?
(A) Because in SMT a theory solver commonly only addresses a small subset of the total constraints.

How good is this approach?

For problems where the solution is SAT this approach tends to determine the solution **much faster** than CAD alone as it can terminate earlier when a satisfying witness is discovered.

For UNSAT problems this approach can still be faster if it allows to reach the conclusion by studying multiple smaller problems; but it may still require the computation of some very large decompositions.

How to adapt CAD further to avoid this?
Try to follow the success of SAT-solvers which search the sample spaces by: making guesses, propagating, and generalising conflicts to avoid similar parts of the search space.

Conflict Driven Cylindrical Algebraic Covering (Slide 50/58)

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

Features:

- Search based: choose sample point and if not satisfying build cell around it using CAD technology.
- Cells form covering of $\mathbb{R}^n$ instead of decomposition.
  Can use far fewer cells!
- Cells are still arranged in cylinders making projection easy.
- Conflict Driven search guides away from past conflicts.

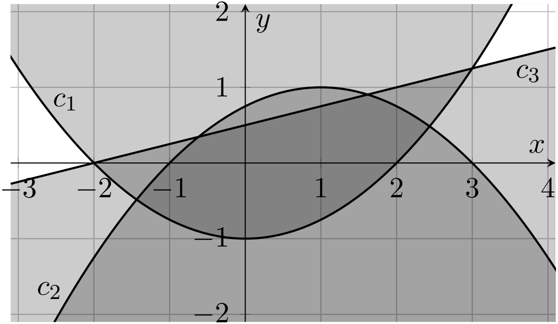
CDCAC: Basic Idea

- Pick sample for lowest variable in ordering.
- Extend to increasingly higher dimensions in reference to those constraints made univariate.
- If all constraints satisfied then conclude SAT.
- If a constraint cannot be satisfied generalise to CAD cell in current dimension.
- Search outside the cell in that dimension.
- If entire dimension covered by cells then generalise to rule out cell in dimension below using CAD projection.
- Conclude UNSAT when covering for lowest dimension obtained.

Following slides by Gereon Kremer show simple example.

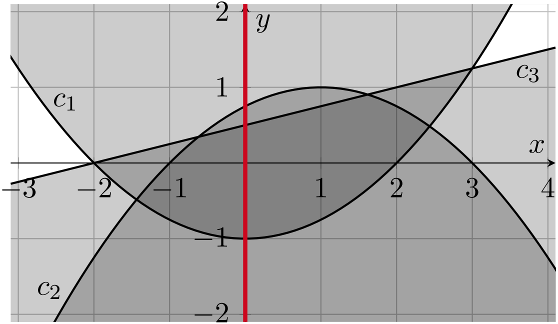
An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad$ $c_2 : 4 \cdot y > 4 - (x-1)^2$ $\quad$ $c_3 : 4 \cdot y > x + 2$

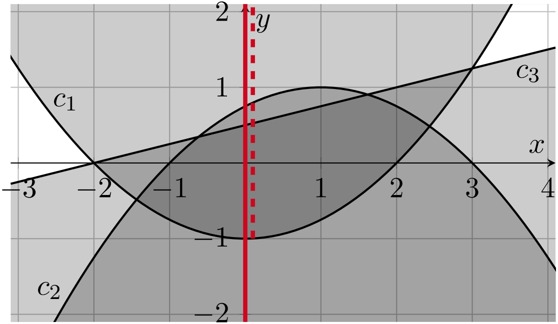


No constraint for x

Guess $x \mapsto 0$

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



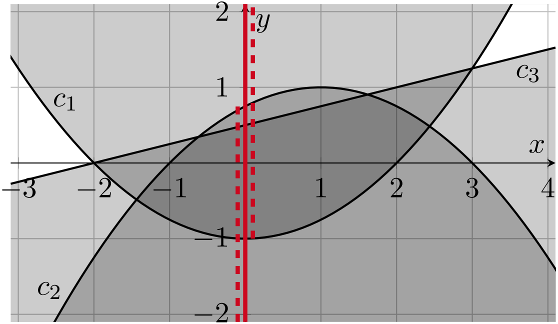
No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

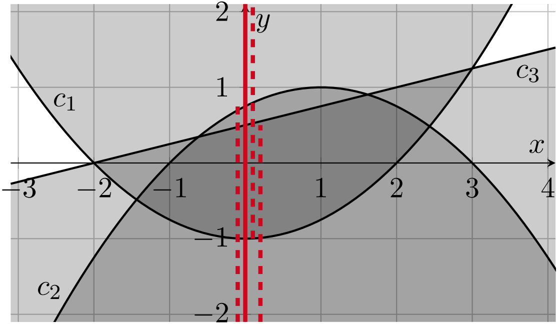
$c_1 \rightarrow y \notin (-1, \infty)$

$c_2 \rightarrow y \notin (-\infty, 0.75)$

RWTH AACHEN UNIVERSITY

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad c_2 : 4 \cdot y > 4 - (x-1)^2$ $\quad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

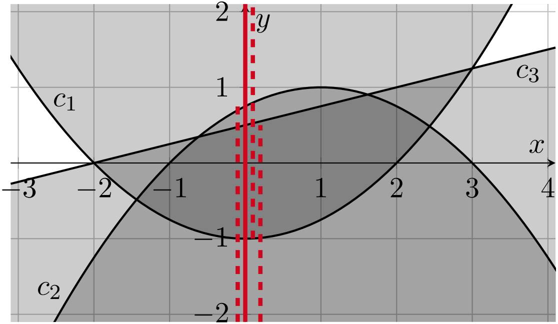
$c_2 \to y \notin (-\infty, 0.75)$

$c_3 \to y \notin (-\infty, 0.5)$

RWTH AACHEN UNIVERSITY

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

$c_2 \to y \notin (-\infty, 0.75)$

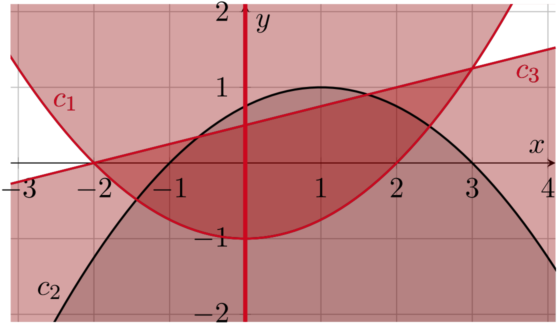
$c_3 \to y \notin (-\infty, 0.5)$

Construct covering

$(-\infty, 0.5), (-1, \infty)$

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad c_2 : 4 \cdot y > 4 - (x-1)^2$ $\quad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

$c_2 \to y \notin (-\infty, 0.75)$

$c_3 \to y \notin (-\infty, 0.5)$

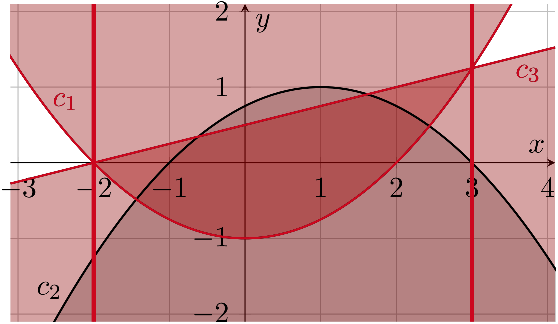
Construct covering

$(-\infty, 0.5), (-1, \infty)$

RWTH AACHEN UNIVERSITY

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

$c_1 \rightarrow y \notin (-1, \infty)$

$c_2 \rightarrow y \notin (-\infty, 0.75)$

$c_3 \rightarrow y \notin (-\infty, 0.5)$

Construct covering

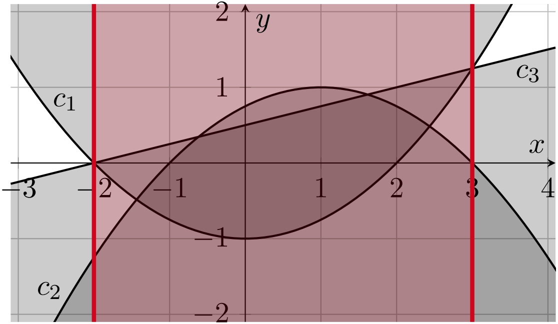
$(-\infty, 0.5), (-1, \infty)$

Construct interval for x

$x \notin (-2, 3)$

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

$c_2 \to y \notin (-\infty, 0.75)$

$c_3 \to y \notin (-\infty, 0.5)$

Construct covering

$(-\infty, 0.5), (-1, \infty)$

Construct interval for x

$x \notin (-2, 3)$

New guess for x

CDCAC in cvc5

cvc5 is a popular open-source SMT-solver used by prominently in academia and industry (e.g. Amazon Web Services). It won the 2022 SMT Competition overall, and the QFNRA track. cvc5 used CDCAC as the core algorithm for non-linear real arithmetic.

G. Kremer, A. Reynolds, C. Barrett, and C. Tinelli.
Cooperating techniques for solving nonlinear real arithmetic in the cvc5 SMT solver (system description),
In: Automated Reasoning, (LNCS 13385), pages 95-105.
Springer International Publishing, 2022.

Outline

Symbolic Computation VS Machine Learning

Machine Learning (ML) can use statistics and big data to learn how to perform tasks that have not been explicitly programmed.

(Q) So can ML replace symbolic computation?

There is a growing body of research on the use of ML in place of expensive symbolic computation. E.g. for the solution of differential equations.

However, like many ML applications, there are issues of over-fitting: where the ML does very well on the data used to train it but poorly on different data. Unlike other ML applications it is not always obvious which data is similar to the training data.

ODE solving is well suited because it is cheap to symbolically check the correctness of the answer. This is not the case for most symbolic computation.

Symbolic Computation WITH Machine Learning (Slide 54/58)

ML can only offer probabilistic guidance, but symbolic computation prizes exact results. 99% accuracy is great for image recognition but would not be acceptable for a mathematical proof.

However: ML can be applied to symbolic computation and still ensure exact results; by having it guide existing algorithms rather than replace them entirely.

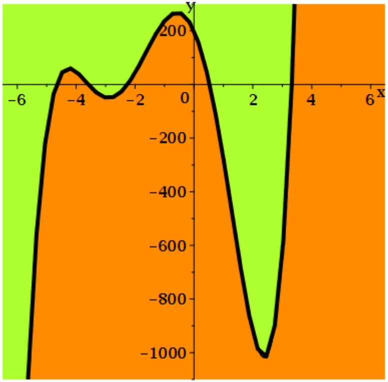
Symbolic Computation algorithms will often come with choices that need to be made but which do not effect the mathematical correctness of the final result; but do effect the resources required to find that result, and how the result is presented.

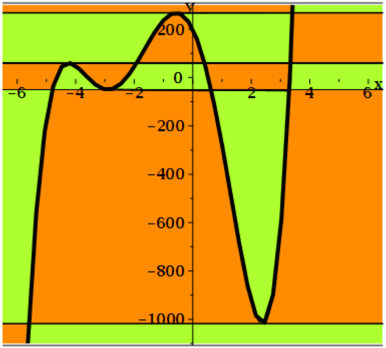
Such choices are often either left to the user, hard coded by the developer, or made based on a simple heuristic. ML may be able to offer a superior choice.

Example: Variable Ordering for Quantifier Elimination

(Slide 55/58)

CAD requires a variable ordering: there can be multiple valid orderings which lead to an acceptable decomposition, but some lead to smaller decompositions via less computation.





This is INTERESTING Machine Learning

So ML has great potential for Symbolic Computation. But note this is also a particularly challenging / interesting ML domain:

- No a priori limit on the input space.
- Supervised learning hard: because labelling dataset needs lots of expensive symbolic computation.
- Unsupervised learning is hard: because it is unclear if a particular outcome is good or bad without seeing the competition!
- What constitutes a meaningful and representative data set?
- Insufficient quantities of real world data for deep learning.
- How to perform data augmentation / synthetic data generation to allow for good generalisability on problems of interest?

Beyond Efficiency Gains

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 a simple, "*human level*" strategy that explained most of the choices. Simple, but very different the human-designed heuristics.

Suggests that ML can reveal new mathematical truths about the algorithm, and direct future non-ML algorithm development. But how to automate the analysis?

Recent work on XAI for QE variable ordering (Slide 58/58)

L. Pickering, T. Del Rio Almajano, M. England and K. Cohen.
Explainable AI Insights for Symbolic Computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition.
In Press: Journal of Symbolic Computation, 2023.

We applied the SHAP XAI tool to analyse the features used in a CAD ML-based variable ordering classifier. Some identified as impactful had been known before but others were new.

Next constructed non-ML heuristics from trio's of these features, in a similar design to prior human-designed heuristics. The best of these outperform's the state-of-the-art on standard benchmarks.

Demonstrates the potential for XAI as an exploratory tool in human optimisation of symbolic computation algorithms.

Contact Details

Matthew.England@coventry.ac.uk

https://matthewengland.coventry.domains/

Thanks for Listening