The Computer Algebra view of "solving"

James Davenport
Hebron & Medlock Professor of Information Technology[1]

University of Bath (U.K.)

31 August 2016

[1] Thanks to Matthew England and SC^2: H2020-FETOPEN-2016-2017-CSA project 712689: www.sc-square.org

Concepts

Given a set of polynomial equations in $k[x_1, \ldots, x_n]$, how do we solve them

* Or possibly "describe the solutions" if infinitely many

1. Gröbner bases — solves over $\overline{k}$
2. Regular Chains — solves over $\overline{k}$
3. Cylindrical Algebraic Decomposition — solves over $\mathbf{R}$

SMT Of course, we miight only want one solution, or the existence of a solution

Base case

Given A set of linear equations

Reduce to triangular form

$$\begin{pmatrix} 1 & ? & ? & \dots & ? \\ 0 & 1 & ? & \dots & ? \\ 0 & 0 & 1 & \dots & ? \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Solve by back substitution: x_n is obvious, then x_{n-1} is obvious, and so on

Implicitly We've imposed an order on the variables

Nonlinear equations: Order

Ordering the variables is not enough: does x_1^2 come before x_1x_2? Before $x_1x_2^2$? etc.

Lexicographic Sort on x_1 powers, then on x_2 powers ...

Degree lex Sort on total degrees first, then break ties by lex

$$x^3 > x^2y > x^2z > xy^2 > xyz > xz^2 > y^3 > y^2z > yz^2 > z^3$$

Degrevlex Sort on total degrees, then break ties by reverse lex

$$x^3 > x^2y > xy^2 > y^3 > x^2z > xyz > y^2z > xz^2 > yz^2 > z^3$$

Elimination Something on $x_1, \ldots, x_k$, breaking ties on $x_{k+1}, \ldots, x_n$

In general, lexicographic is the most useful, but degrevlex the fastest to compute (but see [vH15])

Beyond Gaussian Elimination

Gaussian can still be done

Also Reduction (division): $x_1x_2 + x_2$ reduces $x_1^2x_2 + x_3$ to $-x_1x_2 + x_3$, which reduces to $x_2 + x_3$ (and this reduces $x_1x_2 + x_2$ to $-x_1x_3 - x_3$)

Insufficient What about $f := x_1^2x_2 + x_2x_3^2$ and $g := x_1x_2^2 + x_4$?

$S(f,g)$ $:= x_2f - x_1g = \cancel{x_1^2}\,\cancel{x_2^2} + x_2^2x_3^2 - (\cancel{x_1^2}\,\cancel{x_2^2} + x_1x_4) = x_2^2x_3^2 - x_1x_4 =: h$

$S(g,h)$ $:= x_3^2g - x_1h = \cancel{x_1}\,\cancel{x_2^2}\,\cancel{x_3^2} + x_3^2x_4 - (\cancel{x_1}\,\cancel{x_2^2}\,/\,x_3^2 - x_1^2x_4) = x_1^2x_4 + x_3x_4$

Note The degree has grown, nevertheless [Buc65] this process terminates, in a *Gröbner basis*

$$[{x_2}^4{x_3}^2 + {x_4}^2, -{x_2}^2{x_3}^2 + x_1\,x_4, x_1\,{x_2}^2 + x_4, {x_1}^2x_2 + x_2\,{x_3}^2]$$

Lex Gröbner bases look like (finitely many solutions)

Generally (Shape Lemma [BMMT94])

$$\begin{pmatrix} x_1 & 0 & 0 & \dots & p_1(x_n) \\ 0 & x_2 & 0 & \dots & p_2(x_n) \\ 0 & 0 & x_3 & \dots & p_3(x_n) \\ \dots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & p_n(x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Solve by back-substitution

But not always this shape, e.g.

3 points $\{x_1^2 - 1, x_1(x_2 - 1) - x_2 + 1, x_2^2 - 1\}$

[Gia89, Kal89] Intelligent back-substitution can still work

Also Can convert to Lex by [FGLM93]

Regular Chains/Triangular Decompositions

Regular Chain $T := (f_1, \ldots, f_k)$ such that the f_i have distinct main variables, and each $\mathrm{lt}_{\mathrm{mvar}(f_i)}(f_i)$ is invertible with respect to T .

3 points $\left\{x_1^2 - 1, x_1(x_2 - 1) - x_2 + 1, x_2^2 - 1\right\}$ is not a regular chain

But RCs $\left\{x_1^2 - 1, x_2 - 1\right\}$ and $\{x_1(x_2 - 1) - x_2 + 1, x_2 + 1\}$

Quasivariety $W(T) = V(T) \setminus V\left(\prod \mathrm{lc}_{\mathrm{mvar}(f_i)}(f_i)\right)$: those things that are proved zero by T, without "suspicious cancellation"

(Lazard) **Triangular Decomposition** Produce a set of Regular Chains T_i from F such that $V(F) = \bigcup W(T_i)$

Quantifier Elimination

Throughout, $Q_i \in \{\exists, \forall\}$. Given

$$\Phi := Q_{k+1}x_{k+1} \ldots Q_n x_n \phi(x_1, \ldots, x_n),$$

where ϕ is in some (quantifier-free, generally Boolean-valued) language L, produce an equivalent

$$\Psi := \psi(x_1, \ldots, x_k) : \qquad \psi \in L$$

In particular, $k = 0$ is a decision problem: is Φ true?

Quantifier Elimination is difficult

$$\forall n : n > 1 \Rightarrow \exists p_1 \exists p_2 \left(p_1 \in \mathcal{P} \wedge p_2 \in \mathcal{P} \wedge 2n = p_1 + p_2\right)$$

$$\left[m \in \mathcal{P} \equiv \forall p \forall q \left(m = pq \Rightarrow p = 1 \vee q = 1\right)\right]$$

is a statement of Goldbach's conjecture with, naïvely, seven quantifiers (five will do)

In fact, quantifier elimination is impossible over **N**. [Mat70]

However, it is possible for semi-algebraic (polynomials and inequalities) L over **R** [Tar51]

Unfortunately, the complexity of Tarski's method is indescribable

Over $\mathbf{R}$ we can add $>$ to $=$

(must) $\exists y : y^2 = x \Leftrightarrow x \geq 0$

Hence Semi-algebraic geometry, or real algebraic geometry

CAD "Cylindrical (semi-)Algebraic Decomposition": A partition of $\mathbf{R}^n$ into semi-algebraic sets D_i such that $\forall i, j, k$, if $(x_1, \ldots, x_n) \mapsto_\pi (x_1, \ldots, x_k)$, **either** $\pi(D_i) = \pi(D_j)$ **or** $\pi(D_1) \cap \pi(D_j) = \emptyset$

Also Each D_i has a sample point α_i

Given set f_i of polynomials, construct a CAD *sign-invariant* for every f_i

from a CAD we can read off the answer to any QE problem (quantified in $x_l, \ldots, x_n$ in that order)

Collins' method [Col75]

1. Let $\mathcal{S}_n$ be the polynomials in ϕ (m polynomials, degree d, n variables)
2. Compute $\mathcal{S}_{n-1}$ ($\Theta(m^2)$ polys, degree $\Theta(d^2)$, $n-1$ variables)
3. and $\mathcal{S}_{n-2}$ ($\Theta((m^2)^2)$ polys, degree $\Theta((d^2)^2)$, $n-2$ variables)

$\vdots$ continue

n and $\mathcal{S}_1$ ($\Theta(m^{2^{n-1}})$ polys, degree $\Theta(d^{2^{n-1}})$, 1 variable)

$n+1$ Isolate roots of S_1

$n+2$ Over each root, or interval between roots, isolate roots of S_2

$\vdots$ continue

$2n$ S_n has invariant signs on each region of $\mathbf{R}^n$, so $\phi(x_1,\ldots,x_n)$ has invariant truth on each region

$2n+1$ So evaluate truth of Φ on each region of $(x_1,\ldots,x_k)$-space

Clearly complexity $(md)^{2^{O(n)}}$: in fact $O\left((2m)^{2^{2n+8}}d^{2^{n+6}}\right)$ [Col75]

Collins' method continued

Well, at least that's describable, even if worrying

A better analysis of step $n+1$ [Dav85] gives $O\left((2k)^{2^{2n+\cancel{8}6}} d^{2^{n+\cancel{6}4}}\right)$

which doesn't look very impressive until you realise it's $Z^4 \to Z$

In fact, it largely affects the analysis, not the actual running time

[DH88] showed QE is $\Omega\left(2^{2^{(n-2)/6}}\right)$, or (harder) $\Omega\left(2^{2^{(n-2)/5}}\right)$

(at least in the dense model, i.e. storing all $d+1$ coefficients of a polynomial of degree d).

So we're in $\left(2^{2^{\Theta(n)}}\right)$-land:

this is not the same as $\Theta\left(2^{2^n}\right)$-land, of course

More lower bounds [BD07]

The key idea [Hei83]: suppose Φ_n is $y_n = f_n(x_n)$. Then

$$\Phi_{n+1}(x_{n+1}, y_{n+1}) := \exists z_n \forall x_n \forall y_n$$
$$[(y_n = y_{n+1} \wedge x_n = z_{n+1}) \vee (y_n = z_{n+1} \wedge x_n = x_{n+1})] \Rightarrow \Phi_n(x_n, y_n)$$

is $y_{n+1} = f_n(f_n(x_{n+1}))$. Apply this to

$$f_0(x_0) = \begin{cases} 2x & x \leq 1/2 \\ 2 - 2x & x > 1/2 \end{cases}$$

Then $\Phi_n(x_n, \frac{1}{2})$ defines a set with 2^{2^n} isolated points.
[BD07] shows this set needs doubly exponential space to encode, in dense, sparse or factored form.
However each solution itself is at most singly-exponential ([DH88] has individual solutions doubly-exponential)

Changing the Question

The Heintz construction of [BD07] is $\underbrace{\exists\forall\forall}_{\text{block}} \cdots \underbrace{\exists\forall\forall}_{\text{block}}$, with two **alternations** of quantifiers for every three quantifiers

Let a be the number of alternations

Then [FGM90] the (sequential) cost is $(md)^{n^{O(a)}}$

The doubly-exponential nature is really only for the number of alternations, and it's singly-exponential for the number of variables

I know of no implementation of this method

But It means that cylindrical algebraic decomposition is not always (asymptotically!) best

Order is (sometimes) everything

Consider the polynomial [BD07, Theorem 7]

$$\left((y_{n-1}-\frac{1}{2})^2+(x_{n-1}-z_n)^2\right)\left((y_{n-1}-z_n)^2+(x_{n-1}-x_n)^2\right)x^{n+1}$$
$$+\sum_{i=1}^{n-1}\left((y_{i-1}-y_i)^2+(x_{i-1}-z_i)^2\right)\left((y_{i-1}-z_i)^2+(x_{i-1}-x_i)^2\right)x^{i+1}$$
$$+\left((y_0-2x_0)^2+(\alpha^2+(x_0-\frac{1}{2}))^2\right)\times$$
$$\left((y_0-2+2x_0)^2+(\alpha^2-(x_0-\frac{1}{2}))^2\right)x+a$$

Eliminating $a, x_n, z_n, x_{n-1}, y_{n-1}, z_{n-1}\ldots, z_1, x_0, \alpha, y_0, x$ gives a CAD (in fact a polynomial in a) with at least 2^{2^n} cells, whereas the opposite order has three cells.

Conversely [BD07, Theorem 8] there are problems that are doubly exponential for all orders.

If we can choose the order, how?

Various heuristics:

- **sotd** For all $n!$ orders, perform steps 1-n, measure sotd (sum of total degrees) and do $n+1,\ldots$ for the least
- **Greedy sotd** [DSS04] Do step 1 for each variable, choose the best (sotd) and repeat: often ties
- **ndrr** [BDEW13] For all $n!$ orders, perform steps 1-n, count number of distinct real roots
- **we** tend to use greedy sotd with ndrr as a tiebreaker
- **Brown** [Bro04, 5.2] Eliminate lowest degree variable first (with tie-breaking rules): quite effective

Machine Learning metaheuristic: results from [HEW$^+$14] are encouraging (but what's the benchmark?)

Ordering Example [DSS04]

Lazard's quartic: $\forall x : px^2 + qx + r + x^4 \geq 0$
6 possible orders for (p, q, r)

order	sotd	#cells	CAD	#true	QE
1	54	445	4.71	251	7.04
2	54	445	83.39	251	138.18
3	50	417	0.54	235	0.89
4	50	417	1.64	239	2.55
5	66	—	>600	—	>600
6	66	—	>600	—	>600

Equational Constraints [McC99]

If ϕ is $f = 0 \wedge \hat{\phi}$, we need only consider the cells when $f = 0$ is true. This means the first projection step produces $O(m)$ polynomials rather than $O(m^2)$, and the complexity is
$O\left((2m)^{2^{2n+\cancel{8}6}} d^{2^{n+6}}\right)$.
This gives an interesting formulation problem: given

$$(f_1 = 0 \wedge g_1 < 0) \vee (f_2 = 0 \wedge g_2 < 0) \qquad (1)$$

we are better off solving the equivalent

$$f_1 f_2 = 0 \wedge [(f_1 = 0 \wedge g_1 < 0) \vee (f_2 = 0 \wedge g_2 < 0)] \qquad (2)$$

even though the degree goes up: $O\left((2m)^{2^{2n+\cancel{8}6}} d^{2^{n+\cancel{6}7}}\right)$

[There is a technical side-condition *well-orientedness*, possibly obsoleted [MPP16]]

Truth-Table invariant CAD [BDE+16]

In

$$(f_1 = 0 \wedge g_1 < 0) \vee (f_2 = 0 \wedge g_2 < 0) \tag{3}$$

the first projection set need only be $\mathrm{Disc}(f_1)$, $\mathrm{Disc}(f_2)$, $\mathrm{Res}(f_1, f_2)$, $\mathrm{Res}(f_1, g_1)$, $\mathrm{Res}(f_2, g_2)$ (and omits $\mathrm{Disc}(g_1)$, $\mathrm{Disc}(g_2)$, $\mathrm{Res}(g_1, g_2)$, $\mathrm{Res}(f_1, g_2)$, $\mathrm{Res}(f_1, g_2)$). Essentially all the advantages of equational constraints.

There is still the technical side-condition *well-orientedness*, removed (with many other improvements) in [BCD+14]

There are still issues of formulation: e.g. in $(f_1 = 0 \wedge f_2 = 0 \wedge g_1 < 0) \vee \ldots$, which equation do we prefer?

Alternative method: CAD by Regular Chains [CM14]

C Compute a triangular decomposition over **C**

Hence different challenging problems (may) live in different decompositions

Then Make it *semi-algebraic*, i.e. work out where real lines cross.

Note That this is where different problems interact

Then construct the CAD

Choice of Equational Constraint [BDE+16]

EC Choice 1				EC Choice 2				EC Choice 3			
Cells	Time	S	N	Cells	Time	S	N	Cells	Time	S	N
657	5.6	61	7	463	5.1	64	8	269	1.3	**42**	**4**
711	6.3	66	6	471	5.4	71	6	303	1.1	**40**	**5**
375	2.7	81	9	435	3.6	**73**	**8**	425	2.8	80	**8**
1295	21.4	140	13	477	3.8	**84**	**9**	1437	23.9	158	14
285	2.0	61	7	169	1.0	**59**	**5**				
39	0.1	54	5	9	0.0	**47**	**1**				
F	-	**14**	**0**	F	-	**14**	**0**	27	0.1	**14**	**0**
57	0.3	**32**	**3**	117	0.7	35	**3**	119	0.6	36	4

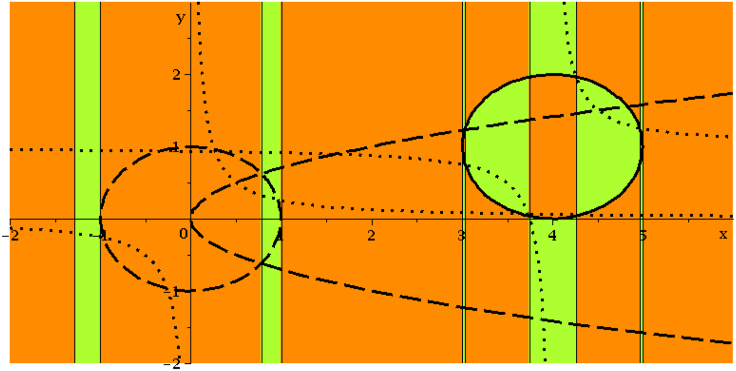
Table: Comparing the choice of equational constraint for a selection of problems. The lowest cell count for each problem is highlighted and the minimal values of the heuristics emboldened.

Which constraint?

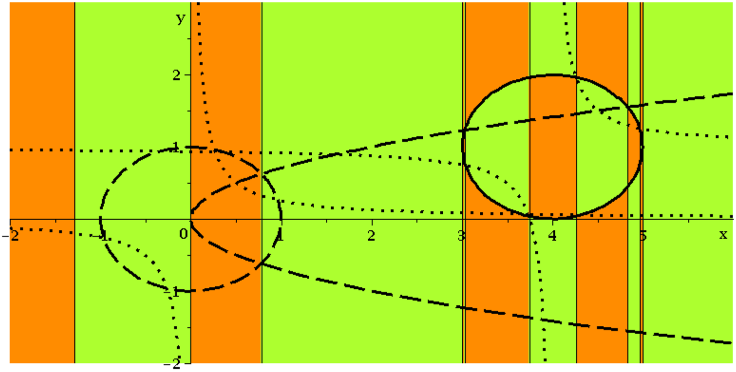
We assume $x \prec y$ and consider $\{\phi_1, \phi_2\}$:

$$\begin{aligned} f_1 &:= x^2 + y^2 - 1, \qquad h := y^2 - \tfrac{x}{2}, \qquad g_1 := xy - \tfrac{1}{4} \\ f_2 &:= (x-4)^2 + (y-1)^2 - 1 \quad g_2 := (x-4)(y-1) - \tfrac{1}{4}, \\ \phi_1 &:= h = 0 \wedge f_1 = 0 \wedge g_1 < 0,\ \phi_2 := f_2 = 0 \wedge g_2 < 0. \end{aligned} \tag{1}$$

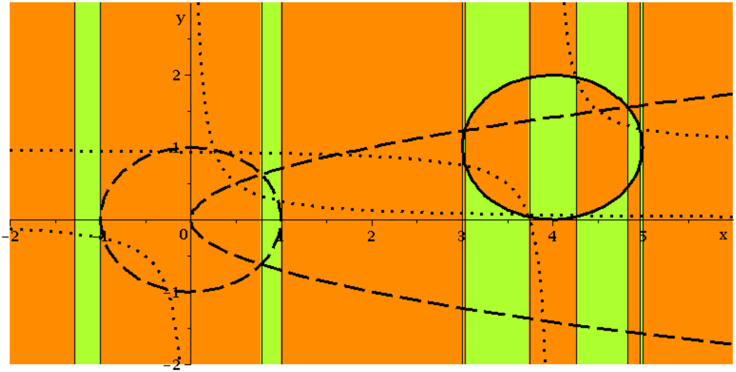
RC-TTICAD with $f_1 \to h \to f_2$ (57 cells).



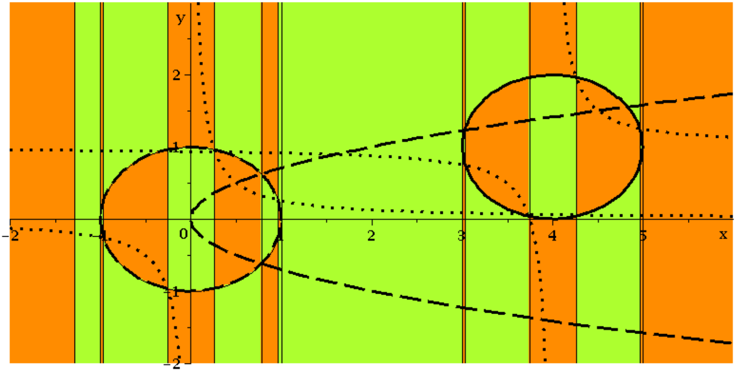
RC-TTICAD with $h \rightarrow f_1 \rightarrow f_2$ (75 cells). This is the default and the same as with f_2, h, f_1.



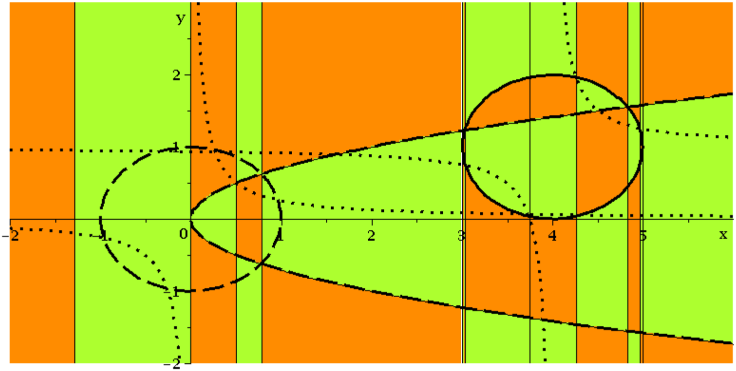
RC-TTICAD with $f_2 \to f_1 \to h$ (77 cells).



PL-TTICAD with f_1 identified (117 cells).



RC-TTICAD with *h* identified (163 cells).



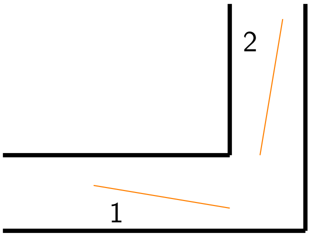
Gröbner Reduction as well [BDEW13]

Order	Full CAD		TTI CAD					TTI+Grö CAD				
	Cells	Time	Eq Const	Cells	Time	S	N	Eq Const	Cells	Time	S	N
$y \prec x$	725	22.802	$f_{1,1}, f_{2,1}$	153	0.818	**62**	12	$\hat{f}_{1,1}, \hat{f}_{2,1}$	27	0.095	**37**	**3**
			$f_{1,1}, f_{2,2}$	111	0.752	94	10	$\hat{f}_{1,1}, \hat{f}_{2,2}$	47	0.361	50	5
			$f_{1,2}, f_{2,1}$	121	0.732	85	9	$\hat{f}_{1,1}, \hat{f}_{2,3}$	93	0.257	50	9
			$f_{1,2}, f_{2,2}$	75	0.840	99	**7**	$\hat{f}_{1,2}, \hat{f}_{2,1}$	47	0.151	47	5
								$\hat{f}_{1,2}, \hat{f}_{2,2}$	83	0.329	63	7
								$\hat{f}_{1,2}, \hat{f}_{2,3}$	145	0.768	81	11
								$\hat{f}_{1,3}, \hat{f}_{2,1}$	95	0.263	46	10
								$\hat{f}_{1,3}, \hat{f}_{2,2}$	151	0.712	80	12
								$\hat{f}_{1,3}, \hat{f}_{2,3}$	209	0.980	62	16
$x \prec y$	657	22.029	$f_{1,1}, f_{2,1}$	125	0.676	**65**	14	$\hat{f}_{1,1}, \hat{f}_{2,1}$	29	0.085	**39**	**4**
			$f_{1,1}, f_{2,2}$	117	0.792	96	11	$\hat{f}_{1,1}, \hat{f}_{2,2}$	53	0.144	52	6
			$f_{1,2}, f_{2,1}$	117	0.728	88	11	$\hat{f}_{1,1}, \hat{f}_{2,3}$	97	0.307	53	97
			$f_{1,2}, f_{2,2}$	85	0.650	101	**8**	$\hat{f}_{1,2}, \hat{f}_{2,1}$	53	0.146	49	6
								$\hat{f}_{1,2}, \hat{f}_{2,2}$	93	0.332	65	8
								$\hat{f}_{1,2}, \hat{f}_{2,3}$	149	0.782	81	13
								$\hat{f}_{1,3}, \hat{f}_{2,1}$	97	0.248	48	11
								$\hat{f}_{1,3}, \hat{f}_{2,2}$	149	0.798	82	13
								$\hat{f}_{1,3}, \hat{f}_{2,3}$	165	1.061	65	18

Table 2. Experimental results relating to Example 7. The lowest cell counts are high

Robot Motion Planning

Reduces to CAD [SS83]. But can we move ladder 1 to position 2?



Insoluble in 1986 [Dav86], insoluble today by [SS83, and today's hardware and CAD advances]

A different formulation [WDEB13]

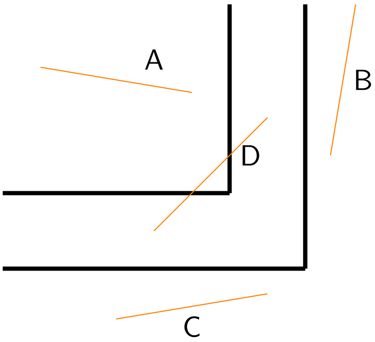


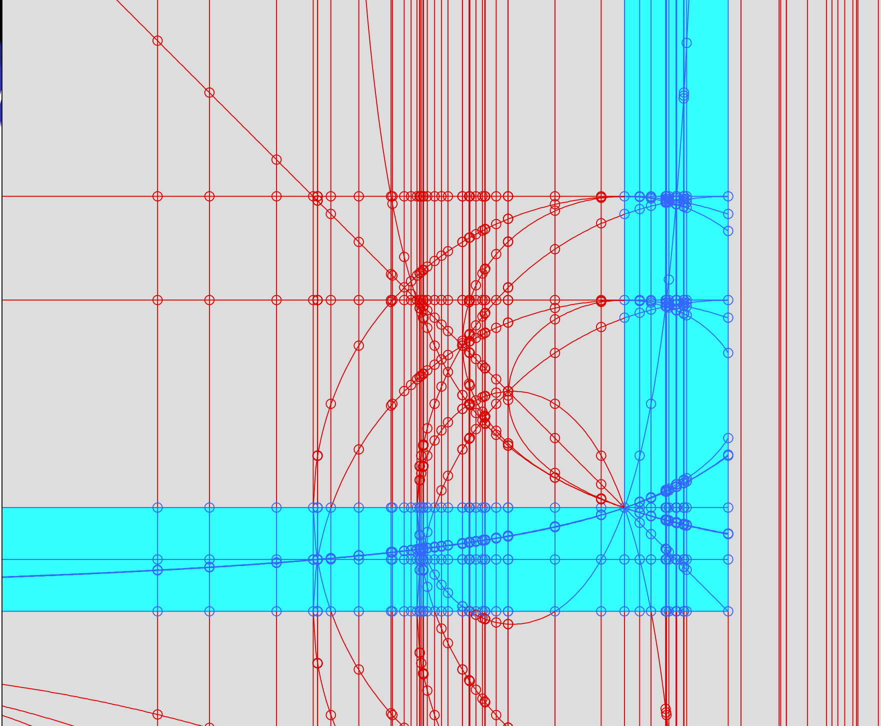
Figure: Four canonical invalid positions of the ladder. Note from the algebraic descriptions that for positions A–C only one end need be outside the corridor.

length$\wedge\neg(A \vee B \vee C \vee D)$: Soluble (5 hours CPU, 285419 cells)

The solution: (but what does it mean?)

$$
\begin{aligned}
& x \leq 0 \wedge y \geq 0 \wedge w \leq 0 \wedge z \geq 0 \wedge (y-z)^2 + (x-w)^2 = 9 \\
& \wedge \Big[[x+1 \geq 0 \wedge w+1 \geq 0] \vee \big[y-1 \leq 0 \wedge w+1 \geq 0 \\
& \qquad \wedge\, y^2w^2 - 2yw^2 + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w \\
& \qquad\quad + 4xyw - 2x^3w - 4x^2w - 4xw + x^2y^2 - 2x^2y \\
& \qquad\quad + x^4 + 2x^3 - 7x^2 - 18x - 9 \geq 0 \big] \\
& \quad \vee \big[x+1 \geq 0 \wedge yw - w + y + x \geq 0 \wedge w^2 - 2xw + y^2 \\
& \qquad - 2y + x^2 - 8 > 0 \wedge z - 1 \leq 0 \big] \\
& \quad \vee \big[x+1 \geq 0 \wedge yw - w + y + x \geq 0 \wedge y^2w^2 - 2yw^2 \\
& \qquad + x^2w^2 + 2xw^2 + 2w^2 - 2xy^2w + 4xyw - 2x^3w \\
& \qquad\quad - 4x^2w - 4xw + x^2y^2 - 2x^2y + x^4 + 2x^3 - 7x^2 \\
& \qquad\quad - 18x - 9 \leq 0 \wedge z - 1 \leq 0 \big] \\
& \quad \vee [y - 1 \leq 0 \wedge z - 1 \leq 0] \Big].
\end{aligned} \tag{4}
$$

How



Conclusions

The more I learn, the less I know, but

- There's more than one way to state a problem
- Clearly equivalent in terms of **decidability**, but not **practical computability**
- The differences are vast in practice
- We have some reasonable heuristics
- But much more work needs to be done, theoretically, experimentally, and on the "software packaging" side
- We need practical work on alternative methods for quantifier elimination

Bibliography I

R.J. Bradford, C. Chen, J.H. Davenport, M. England, M. Moreno Maza, and D.J. Wilson.
Truth Table Invariant Cylindrical Algebraic Decomposition by Regular Chains.
In *Proceedings CASC 2014*, pages 44–58, 2014.

C.W. Brown and J.H. Davenport.
The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition.
In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

R.J. Bradford, J.H. Davenport, M. England, S. McCallum, and D.J. Wilson.
Truth table invariant cylindrical algebraic decomposition.
J. Symbolic Computation, 76:1–35, 2016.

Bibliography II

R.J. Bradford, J.H. Davenport, M. England, and D.J. Wilson.
Optimising Problem Formulation for Cylindrical Algebraic Decomposition.
In J. Carette *et al.*, editor, *Proceedings CICM 2013*, pages 19–34, 2013.

E. Becker, M.G. Marinari, T. Mora, and C. Traverso.
The shape of the shape lemma.
In *Proceedings ISSAC 1994*, pages 129–133, 1994.

C.W. Brown.
Tutorial handout.
`http://www.cs.usna.edu/~wcbrown/research/ISSAC04/handout.pdf`, 2004.

Bibliography III

B. Buchberger.
Ein Algorithmus zum Auffinden des basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.
PhD thesis, Math. Inst. University of Innsbruck, 1965.

C. Chen and M. Moreno Maza.
An Incremental Algorithm for Computing Cylindrical Algebraic Decompositions.
In Ruyong Feng, Wen-shin Lee, and Yosuke Sato, editors, *Computer Mathematics*, pages 199–221. Springer Berlin Heidelberg, 2014.

Bibliography IV

G.E. Collins.
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition.
In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

J.H. Davenport.
Computer Algebra for Cylindrical Algebraic Decomposition.
Technical Report TRITA-NA-8511 NADA KTH Stockholm (Reissued as Bath Computer Science Technical report 88-10), 1985.

J.H. Davenport.
On a "Piano Movers" Problem.
SIGSAM Bulletin 1/2, 20:15–17, 1986.

Bibliography V

J.H. Davenport and J. Heintz.
Real Quantifier Elimination is Doubly Exponential.
J. Symbolic Comp., 5:29–35, 1988.

A. Dolzmann, A. Seidl, and Th. Sturm.
Efficient Projection Orders for CAD.
In J. Gutierrez, editor, *Proceedings ISSAC 2004*, pages 111–118, 2004.

J.C. Faugère, P. Gianni, D. Lazard, and T. Mora.
Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering.
J. Symbolic Comp., 16:329–344, 1993.

Bibliography VI

N. Fitchas, A. Galligo, and J. Morgenstern.
Precise sequential and parallel complexity bounds for the quantifier elimination over algebraic closed fields.
J. Pure and Applied Algebra, 67:1–14, 1990.

P. Gianni.
Properties of Gröbner bases under specializations.
In *Proceedings EUROCAL 87*, pages 293–297, 1989.

J. Heintz.
Definability and Fast Quantifier Elimination in Algebraically Closed Fields.
Theor. Comp. Sci., 24:239–277, 1983.

Bibliography VII

Z. Huang, M. England, D. Wilson, J.H. Davenport, L.C. Paulson, and J. Bridge.
Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition.
In S.M.Watt *et al.*, editor, *Proceedings CICM 2014*, pages 92–107, 2014.

M. Kalkbrener.
Solving systems of algebraic equations by using Gröbner bases.
In *Proceedings EUROCAL 87*, pages 282–292, 1989.

Yu.V. Matiyasevich.
Enumerable sets are Diophantine.
Soviet Math. Doklady 2, 11:354–358, 1970.

Bibliography VIII

S. McCallum.
On Projection in CAD-Based Quantifier Elimination with Equational Constraints.
In S. Dooley, editor, *Proceedings ISSAC '99*, pages 145–149, 1999.

S. McCallum, A. Parusinski, and L. Paunescu.
Validity proof of Lazard's method for CAD construction.
`https://arxiv.org/abs/1607.00264`, 2016.

J.T. Schwartz and M. Sharir.
On the "Piano-Movers" Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds.
Adv. Appl. Math., 4:298–351, 1983.

Bibliography IX

A. Tarski.
A Decision Method for Elementary Algebra and Geometry.
2nd ed., Univ. Cal. Press. Reprinted in *Quantifier Elimination and Cylindrical Algebraic Decomposition* (ed. B.F. Caviness & J.R. Johnson), Springer-Verlag, Wein-New York, 1998, pp. 24–84., 1951.

M. van Hoeij.
Groebner basis in Boolean rings is not polynomial-space.
http://arxiv.org/abs/1502.07220, 2015.

D.J. Wilson, J.H. Davenport, M. England, and R.J. Bradford.
A "Piano Movers" Problem Reformulated.
In *Proceedings SYNASC 2013*, pages 53–60, 2013.