What does “without loss of generality” mean (and how do we detect it)

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

University of Bath (U.K.)

1 August 2016

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

Concepts

1. Given a symmetric formula in a, b, c, the mathematician says "without loss of generality $a \leq b \leq c$"
2. Given a geometric figure in the plane, the mathematician says "without loss of generality P is at $(0,0)$ and Q at $(0,1)$"
3. And there's more general "reasoning by symmetry".

See [Har09] for an excellent treatment of making such proofs formal: pause this talk focuses on *detection*

Possible meanings of w.l.o.g. [Har09]

A: non-degeneracy for example "w.l.o.g. $\alpha \neq 0$", really means "$\alpha = 0$ is a special case, which you can easily see for yourself, so I am not going to bother with it here";

B: exploitation of symmetry as in Schur's inequality $\forall a, b, c \in \mathbf{R}, k \in \mathbf{N}$,

$$0 \leq a^k(a-b)(a-c) + b^k(b-a)(b-c) + c^k(c-a)(c-b),$$

where a typical proof might begin: "Without loss of generality, let $a \leq b \leq c$".

But also C: "$\alpha = 0$ renders the result meaningless, so we shall not consider it further".

1: "without loss of generality $a \leq b \leq c$"

- Works if the formula is invariant under S_n acting on the n variables
- Isn't that a lot of checking?

Proposition

The permutations $(1, 2, \ldots, n)$ *and* $(1, 2)$ *generate* S_n *as a group acting on* $\{1, 2, \ldots, n\}$.

Hence it's sufficient to check that these two permutations leave the formula *mathematically* invariant (*syntactic* invariance is too strong a condition)

Does this help SC^2?

Feeding $0 \le a^2(a-b)(a-c) + b^2(b-a)(b-c) + c^2(c-a)(c-b)$ into Regular Chains [CM14] CAD, we get 31 cells: 14 satisfy $a \le b \le c$, either totally, or, where underlined, only partially

Table: Cells satisfying $a \le b \le c$

$c < 0$	$b < c$	$\underline{\text{all}}$
	$b = c$	$a < c; a = c$
$c = 0$	$b < 0$	$a < b; a = b$
	$b = 0$	$a < 0; a = 0$
$c > 0$	$b < 0$	$\underline{\text{all}}$
	$b = 0$	$\underline{a < c}$
	$0 < b < c$	$\underline{\text{all}}$
	$b = c$	$a < 0; a = 0; 0 < a < c; a = c$

Splitting the "undecided" cells gives us 18/39, again a far cry from the naïve 1/6.

What if it's only a subset of the variables?

Proposition

The permutations $(1,2)$*,* $(1,3)\ldots(1,n)$ *generate* S_n *as a group acting on* $\{1,2,\ldots,n\}$*.*

Hence the obvious greedy algorithm will find as many S_k as act, separately, on the n variables.

2: "without loss of generality P is at . . ."

Depends on the symmetry group acting on (what we guess might be) a geometric configuration

Theorem (Simson's Theorem, [Wan96, Mou16])

Let D be on the circumcircle of the triangle ABC, let P, Q and R be the points of AB, AC and BC where the line to D is perpendicular. Then P, Q and R are collinear.

Let us consider just the first statement "Let D be on the circumcircle of the triangle ABC".

This coordinatises to

$$
\begin{aligned}
x_D{}^2 + y_D{}^2 = {} & \frac{x_D \left(x_A{}^2 y_B - x_A{}^2 y_C - x_B{}^2 y_A + x_B{}^2 y_C + x_C{}^2 y_A - x_C{}^2 y_B + y_A{}^2 y_B - y_A{}^2 y_C - y_A\, y_B{}^2 + y_A\, y_C{}^2 + y_B{}^2 y_C - y_B\, y_C{}^2\right)}{x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B} \\
& + \frac{y_D \left(-x_A \left(x_B{}^2 + y_B{}^2\right) + x_A \left(x_C{}^2 + y_C{}^2\right) + x_A{}^2 \left(x_B - x_C\right) + y_A{}^2 \left(x_B - x_C\right) - x_B \left(x_C{}^2 + y_C{}^2\right) + x_C \left(x_B{}^2 + y_B{}^2\right)\right)}{x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B} + \\
& \frac{1}{4} \frac{\left(x_A{}^2 y_B - x_A{}^2 y_C - x_B{}^2 y_A + x_B{}^2 y_C + x_C{}^2 y_A - x_C{}^2 y_B + y_A{}^2 y_B - y_A{}^2 y_C - y_A\, y_B{}^2 + y_A\, y_C{}^2 + y_B{}^2 y_C - y_B\, y_C{}^2\right)^2}{\left(x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B\right)^2} + \\
& \frac{1}{4} \frac{\left(-x_A \left(x_B{}^2 + y_B{}^2\right) + x_A \left(x_C{}^2 + y_C{}^2\right) + x_A{}^2 (x_B - x_C) + y_A{}^2 (x_B - x_C) - x_B \left(x_C{}^2 + y_C{}^2\right) + x_C \left(x_B{}^2 + y_B{}^2\right)\right)^2}{\left(x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B\right)^2} - \\
& \left(x_A - \frac{1}{2} \frac{x_A{}^2 y_B - x_A{}^2 y_C - x_B{}^2 y_A + x_B{}^2 y_C + x_C{}^2 y_A - x_C{}^2 y_B + y_A{}^2 y_B - y_A{}^2 y_C - y_A\, y_B{}^2 + y_A\, y_C{}^2 + y_B{}^2 y_C - y_B\, y_C{}^2}{x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B}\right)^2 \\
& - \left(y_A + \frac{1}{2} \frac{-x_A \left(x_B{}^2 + y_B{}^2\right) + x_A \left(x_C{}^2 + y_C{}^2\right) + x_A{}^2 \left(x_B - x_C\right) + y_A{}^2 \left(x_B - x_C\right) - x_B \left(x_C{}^2 + y_C{}^2\right) + x_C \left(x_B{}^2 + y_B{}^2\right)}{x_A\, y_B - x_A\, y_C - x_B\, y_A + x_B\, y_C + x_C\, y_A - x_C\, y_B}\right)^2
\end{aligned}
$$

CAS can verify invariance under $z \to z + c$ for all variables, so choose $y_A = 0$

$$
\begin{aligned}
x_D^2 + y_D^2 = {} & \frac{x_D\left(x_A^2 y_B - x_A^2 y_C + x_B^2 y_C - x_C^2 y_B + y_B^2 y_C - y_B\, y_C^2\right)}{x_A\, y_B - x_A\, y_C + x_B\, y_C - x_C\, y_B} - \\
& y_D \frac{y_D - x_A\left(x_B^2 + y_B^2\right) + x_A\left(x_C^2 + y_C^2\right) + x_A^2\left(x_B - x_C\right) - x_B\left(x_C^2 + y_C^2\right) + x_C\left(x_B^2 + y_B^2\right)}{x_A\, y_B - x_A\, y_C + x_B\, y_C - x_C\, y_B} - \\
& \frac{1}{4}\, \frac{\left(x_A^2 y_B - x_A^2 y_C + x_B^2 y_C - x_C^2 y_B + y_B^2 y_C - y_B\, y_C^2\right)^2}{\left(x_A\, y_B - x_A\, y_C + x_B\, y_C - x_C\, y_B\right)^2} - \\
& \left(x_A - \frac{1}{2}\, \frac{x_A^2 y_B - x_A^2 y_C + x_B^2 y_C - x_C^2 y_B + y_B^2 y_C - y_B\, y_C^2}{x_A\, y_B - x_A\, y_C + x_B\, y_C - x_C\, y_B}\right)
\end{aligned}
$$

CAS can verify invariance under $z \to z + c$ for $z \in \{x_A, x_B, x_c, x_D\}$, so choose $x_A = 0$

$$x_D^2 + y_D^2 = \frac{x_D\left(x_B^2 y_C - x_C^2 y_B + y_B^2 y_C - y_B\, y_C^2\right)}{x_B\, y_C - x_C\, y_B} + \frac{y_D\left(-x_B\left(x_C^2 + y_C^2\right) + x_C\left(x_B^2 + y_B^2\right)\right)}{x_B\, y_C - x_C\, y_B}$$

We see dramatic simplification of the formulae.

Rotational Symmetry

In fact, both [Wan96, Mou16] coordinatise with $A = (x_A, 0)$ and $B = (-x_A, 0)$, taking (implicit) advantage of the fact that the problem is invariant under translation (so we can place the midpoint of AB at $(0, 0)$) and rotation (so we can place A and B on the x-axis).

$$x_D{}^2 + y_D{}^2 = \frac{y_D\left(-x_A{}^2 + x_C{}^2 + y_C{}^2\right)}{y_C} + x_A{}^2$$

One further step, which [Wan96, Mou16] could have done, and a computer system could certainly spot, is that the equation is homogeneous, and hence we can pick, say, $x_A = 1$.
However, whilst appearing to be a type B w.l.o.g., exploiting symmetry under dilation, it is also asserting $x_A \neq 0$, thus a type A, or even type C, w.l.o.g. as well.

Does this help SC^2?: the data

Table: CAD of $\mathbf{R}^n$ for numerators of equations

	[CM14]			[McC84, EWBD14]		
Equation	Cells	Time (secs)	Memory MiB	Cells	Time (secs)	Memory MiB
Base	591	4.12	341			
1D trans	591	2.80	235	—	> 9000	
2D trans	591	2.29	188	36531*	807.00	55000
$2D\|_{x_B=1}$	319	3.48	256	30803*	433.20	31460
$2D\|_{x_B=16}$	319	3.53	290			
$2D\|_{x_B=256}$	319	4.24	318			
2D,rot	107	0.47	26	589*	3.89	303
$2D,rot\|_{x_A=1}$	37	0.14	11	245	1.86	108

Timings and memory usage from Maple's `CodeTools[Usage]`, and hence both have (up to) four significant figures.

* Warning that the input is not well-oriented.

Does this help SC^2?: the commentary

[CM14] Spotting the translational symmetry doesn't simplify the result (i.e. the geometry is preserved), but helps somewhat with time/memory.

+ Spotting rotational symmetry definitely helps (fewer cells and some things align vertically)
+ Spotting scaling definitely helps (more by eliminating the degenerate case)

[EWBD14] Translational symmetry seems necessary Why?

+ Rotation is very important
+ Scaling also helps

How might we spot it?'

1D trans Check for invariance under $z \to z + c$ for all variables z simultaneously?

Cheap provided it's *all* the variables; otherwise subsets

2D trans Check (half of) possible subsets of variables for $z \to z + c$ invariance (Call these x_i)

3D trans If there's room check subsets of the rest for $z \to z + c$ invariance

2D rotation For all pairings $x_i, y_{\sigma(i)}$, check invariance under $\forall i (x_i, y_{\sigma(i)}) \to (cx_i - sy_{\sigma(i)}, cy_{\sigma(i)} + sx_i)$ (with $c^2 + s^2 = 1$)

or 3D similarly if we have 3D translational invariance

scaling Obvious way (but what about the degenerate case?)

N.B. we need to know about the translations to deduce the rotations, even if not computationally useful

3: "reasoning by symmetry"

This occurs in [BD07], we construct a formula with $3n + O(1)$ quantifiers defining $S := \left\{ \frac{2k-1}{2^{2^n+1}} : 0 < k < 2^{2^n} \right\}$: each point requires $2^{O(n)}$ bits to express, but there are 2^{2^n} of them

[BD07] assert that an explicit representation of S takes $2^{2^n+O(n)}$ bits, but S is symmetric about $x = \frac{1}{2}$, and that half is symmetric about $x = \frac{1}{4}$ etc., leading in principle to a $2^{O(n)}$-bit representation

Put another way, we don't need to count the solutions individually

there's a better solution to the #SMT problem

This doesn't help (asymptotically) with [DH88], where some solutions require $2^{2^{O(n)}}$ bits to represent, but the ideas might be useful

Conclusions

- It is possible to spot symmetry of the S_n type reasonably cheaply: $O(n^2)$ tests
- Translational symmetry is relatively easy to spot, but per se doesn't seem to help [CM14] CAD much
- However, it's a precursor to spotting rotational symmetry, which is useful
- Scaling is also useful, but we need to worry about the degenerate case

All useful heuristics!

Bibliography I

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.

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.

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

Bibliography II

- M. England, D.J. Wilson, R. Bradford, and J.H. Davenport.
  Using the Regular Chains Library to build cylindrical algebraic decompositions by projecting and lifting.
  In *Proceedings ICMS 2014*, pages 458–465, 2014.
- J. Harrison.
  Without Loss of Generality.
  International Conference on Theorem Proving in Higher Order Logics, pages 43–59, 2009.
- S. McCallum.
  An Improved Projection Operation for Cylindrical Algebraic Decomposition.
  PhD thesis, University of Wisconsin-Madison Computer Science, 1984.

Bibliography III

C. Mou.
Software library for triangular decompositions.
Talk at ICMS 2016, 2016.

D. Wang.
GEOTHER: A geometry theorem prover.
International Conference on Automated Deduction, pages 166–170, 1996.