Séminaire Lotharingien de Combinatoire **65** (2011), Article B65a

HOW TO USE CYLINDRICAL ALGEBRAIC DECOMPOSITION

MANUEL KAUERS *

Abstract. We take some items from a textbook on inequalities and show how to prove them with computer algebra using the Cylindrical Algebraic Decomposition algorithm. This is an example collection for standard applications of this algorithm, intended as a guide for potential users.

1. Introduction

This article does not contain original work. Everything said here has already been said before in one way or another. But there are things that are important enough to be said more than once. I believe that computer algebra tools for handling inequalities are among them. Although meanwhile powerful enough to compete with hand-crafted proofs, computer algebra for inequalities is not yet used as routinely as, for instance, summation algorithms are. This may be because algorithms for inequalities are a bit more difficult to use, or because a certain knowledge of the computational background is necessary to understand why their output is indeed rigorous, or because they have a bad reputation of requiring an unreasonably high amount of memory and computation time.

The purpose of this article is to give an example oriented tutorial for Collins's Cylindrical Algebraic Decomposition (CAD) algorithm [4]. The reader I have in mind is too busy to waste valuable time on doing problems related to polynomial inequalities by hand, wants to know to what extent computer algebra can do such tasks, wants to know which algorithms exist and how they are used, but does not necessarily care what these algorithms do internally. The article, in short, summarizes a part of the invited lecture I gave at the 65-th Séminaire Lotharingien de Combinatoire in Strobl, Austria, September 12–15, 2010. This lecture covered three aspects: (a) basic concepts and standard applications of CAD, (b) computational internals of the CAD algorithm, and (c) applications of CAD to inequalities about recursively defined objects. On the last part, which may be most relevant to combinatorialists, I have already published a survey article [6] describing my own work. This overview is still more or less up to date, only little [10, 7] has happened afterwards in this direction. Also for the second part, there are various good references available with all the details for those who want to write their own code [3, 1]. So it seems sensible to address only the first part here.

Originally, CAD was invented in order to do quantifier elimination over the reals: given a quantified formula, it finds a formula without quantifiers which is equivalent over the

* Supported by the Austrian FWF grant Y464-N18.

reals to the input formula. But it can do more. CAD should really be considered as a general tool for dealing with subsets of $\mathbb{R}^n$ that can be described by polynomial equations and inequalities (semialgebraic sets). The CAD algorithm brings descriptions of such sets into some sort of standardized form from which the answers to a number of otherwise non-trivial questions can be easily extracted. Among many other things, we can

- decide whether or not a given semialgebraic set is empty, finite, open, closed, connected, or bounded;
- decide whether or not a given semialgebraic set is contained in another one;
- determine the (topological) dimension of a given semialgebraic set;
- determine a sample point of a given nonempty semialgebraic set;
- determine the number of points of a given finite semialgebraic set;
- determine a tight bounding box of a given bounded semialgebraic set;
- determine the connected components of a given semialgebraic set;
- determine the boundary, the closure, or the interior of a given semialgebraic set;
- determine the projection of a given semialgebraic set in $\mathbb{R}^n$ to a coordinate subspace $\mathbb{R}^k$ $(k < n)$.

The plan here is to give an example oriented introduction for those who want to learn how to do basic maneuvers with CAD without reading long justifications of technical details that are more relevant for implementors than for users. The goal is to explain what CAD is and how it can be used, rather than how it can be computed.

2. A First Example

Let us start with Item 1.16 of the inequality collection [9]:

$$\sqrt{a^2-b^2}+\sqrt{2ab-b^2}>a \qquad (0<b<a).$$

First of all, there is no need to prove such a statement by hand. CAD can do it for us. For example, using *Mathematica*'s built-in implementation (developed by Adam Strzeboński [11]), it is a matter of typing a single command:

In[1]:= $\mathbf{CylindricalDecomposition[Implies[0<b<a,\sqrt{a^2-b^2}+\sqrt{2ab-b^2}>a],\{a,b\}]}$
Out[1]= True

This result means that the logical statement **Implies[...]** in the input is equivalent to the logical statement True in the output in the sense that, for all real numbers a, b, the former statement is true if and only if the latter statement is true. In other words, the inequality is proven.

The inequality becomes false when the coefficient 2 inside the second square root is replaced by 1. CAD detects this. It does, however, not simply return "false". Instead:

In[2]:= $\mathbf{CylindricalDecomposition[Implies[0<b<a,\sqrt{a^2-b^2}+\sqrt{ab-b^2}>a],\{a,b\}]}$
Out[2]= $a\leq 0\vee\bigl(a>0\wedge\bigl(b<\tfrac{4}{5}a\vee b\geq a\bigr)\bigr)$

This output describes precisely the coordinates of all the points $(a,b) \in \mathbb{R}^2$ for which the implication in the input holds. In other words, we have found

$$\begin{aligned} &\{\,(a,b) \in \mathbb{R}^2 : 0 < b < a \Rightarrow \sqrt{a^2-b^2} + \sqrt{ab-b^2} > a\,\} \\ &= \{\,(a,b) \in \mathbb{R}^2 : a \leq 0 \vee (a > 0 \wedge (b < \tfrac{4}{5}a \vee b \geq a))\,\}. \end{aligned}$$

The latter description can be read as an instruction for constructing points in the set: "First choose a as you like. Then, if $a \leq 0$, then you may also choose b as you like. Otherwise, if $a > 0$, then you may choose b either smaller than $\frac{4}{5}a$ (with the specific a chosen before) or greater than or equal to the chosen a." A possible choice is thus $(a,b) = (1, 1/3)$.

Rather than answering simply the question *whether* a specific formula is true or not, the CAD algorithm determines *where* the formula is true. When it returned True before, it really meant that the formula holds "for all $a,b \in \mathbb{R}$ subject to no restrictions", that is, it holds for all $a,b \in \mathbb{R}$. The formula in the second calculation is true for some a,b, and false for others, and CAD provided us with a description of the region where it holds true. An output False would have meant that there do not exist any points (a,b) at all for which the formula is true.

The precise structure of the output formulas computed by CAD will be described in some more detail in Section 4 below. One of its key features is that formulas with quantifiers (forall, exists) can be turned into quantifier free ones. For example,

In[3]:= **CylindricalDecomposition**[
ForAll[$\{a,b\}$, **Implies**[$0 < b < a$, $\sqrt{a^2-b^2} + \sqrt{ab-b^2} > a$]], {}]

Out[3]= False

asserts that the implication

$$0 < b < a \Rightarrow \sqrt{a^2-b^2} + \sqrt{ab-b^2} > a$$

does *not* hold for all real numbers a,b but suppresses the information on where to find counterexamples.

We can also combine quantified variables with unquantified ("free") variables. We have seen that the inequality at the beginning of this section becomes false when replacing the coefficient 2 inside the second square root by 1. But which numbers can we put in place of this 2 without making the inequality false? The question is easily answered by typing

In[4]:= **CylindricalDecomposition**[
ForAll[$\{a,b\}$, **Implies**[$0 < b < a$, $\sqrt{a^2-b^2} + \sqrt{uab-b^2} > a$]], $\{u\}$]

Out[4]= $u \geq 2$

This result means that the quantified formula **ForAll[...]** in the input is equivalent to the unquantified formula $u \geq 2$ in the output in the sense that for all real numbers u, the former statement is true if and only if the latter statement is true. In other words, the coefficient 2 cannot be improved.

We may allow for more freedom by introducing more parameters. For example, we can extend the range for u to $u \geq 1$ by doing an appropriate scaling on right hand side, as the following calculation tells us:

In[5]:= **CylindricalDecomposition[**
ForAll[{a, b}, Implies[0 < b < a, $\sqrt{a^2 - b^2} + \sqrt{uab - b^2} > va$**]], {u, v}]**

Out[5]= $\left(1 \leq u \leq 2 \land v \leq \sqrt{u-1}\right) \lor (u > 2 \land v \leq 1)$

This means that for any choice $u \geq 1$, there is a choice for v (depending on the chosen u) that makes the inequality valid. For example, choosing $u = \frac{3}{2}$ and $v = \frac{1}{2}\sqrt{2}$ gives the new inequality

$$\sqrt{a^2 - b^2} + \sqrt{\tfrac{3}{2}ab - b^2} > \tfrac{1}{2}\sqrt{2}a \qquad (0 < b < a).$$

Another idea is to restrict the range of b from $(0, a)$ to $(0, va)$ for some constant v. Asking for the possible combinations (u, v) that make the formula valid in the restricted range, we obtain the following.

In[6]:= **CylindricalDecomposition[**
ForAll[{a, b}, Implies[0 < b < va, $\sqrt{a^2 - b^2} + \sqrt{uab - b^2} > a$**]], {u, v}]**

Out[6]= $\left(u \leq -1 \land -1 \leq v \leq 0\right) \lor \left(-1 < u < 0 \land u \leq v \leq 0\right) \lor \left(u = 0 \land v = 0\right)$
$\lor \left(0 < u \leq 2 \land 0 \leq v \leq \frac{4u}{u^2+4}\right) \lor \left(u > 2 \land 0 \leq v \leq 1\right)$

This means that for every choice of u, we can find some v that makes the formula true. For instance, for the specific choice $u = 1/2$ we can take any v in the range from 0 to $4u/(u^2 + 4) = 8/17$ according to the fourth part of the formula. For example, $v = 1/3$ would do.

Output 6 is optimized for finding solutions (u, v) by first deciding on a choice for u and then selecting some v which fits to the chosen u. If we have some specific v in mind and want to find a suitable u for this specific v, then the output is not very helpful. We should redo the computation with v as primary variable and u as secondary variable. In *Mathematica*, this is specified by the second argument of the command **CylindricalDecomposition**. Putting $\{v, u\}$ instead of $\{u, v\}$ there gives

In[7]:= **CylindricalDecomposition[**
ForAll[{a, b}, Implies[0 < b < va, $\sqrt{a^2 - b^2} + \sqrt{uab - b^2} > a$**]], {v, u}]**

Out[7]= $\left(-1 \leq v < 0 \land u \leq v\right) \lor v = 0 \lor \left(0 < v \leq 1 \land u \geq \frac{2}{v} - \frac{2}{v}\sqrt{1-v^2}\right)$

Now it is apparent that we cannot find any u for v with $|v| > 1$. This was not evident from Output 6. Also, we can now easily determine a suitable u for, say $v = \frac{3}{5}$. The third part of Output 7 advises us that the good choices for u are precisely those where $u \geq \frac{2}{3/5} - \frac{2}{3/5}\sqrt{1 - (3/5)^2} = \frac{2}{3}$. We have found the new inequality

$$\sqrt{a^2 - b^2} + \sqrt{\tfrac{2}{3}ab - b^2} > a \qquad (0 < b < \tfrac{3}{5}a).$$

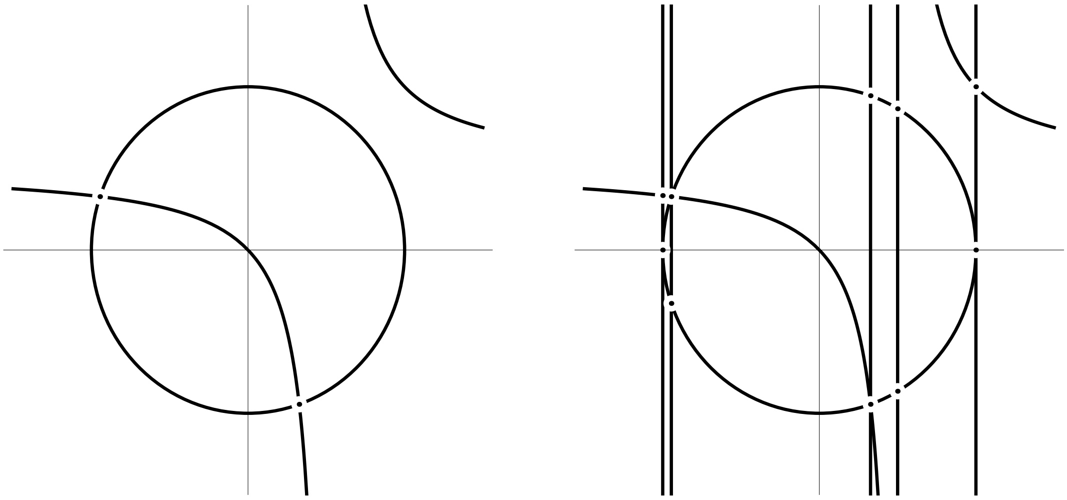

FIGURE 1. a) algebraic decomposition of P b) completion of P to a cylindrical algebraic decomposition

It would be easy to continue producing further variations. The reader shall feel encouraged to discover some interesting ones on her or his own as an exercise.

3. The Geometric Point of View

A finite set $P = \{p_1, \dots, p_m\}$ of polynomials in the variables $x_1, \dots, x_n$ naturally induces a decomposition of $\mathbb{R}^n$ into cells, the so-called *algebraic decomposition.* By a *cell,* we mean here a connected subset of $\mathbb{R}^n$ where all the p_i have the same sign and which is maximal in the sense that making the set larger would necessarily violate sign invariance or connectedness.

As an example, the set $P = \{x^2 + y^2 - 4, (x-1)(y-1) - 1\}$ induces a decomposition of $\mathbb{R}^2$ into 13 cells as depicted in Figure 1a: five "areas", six "arcs" where one of the polynomials is zero but the other is not, and two isolated points on where both polynomials are zero. Note that 0 is considered as a sign of its own: $\mathrm{sgn}(0) := 0$.

If an algebraic decomposition of $\mathbb{R}^n$ is *cylindrical,* then for every $k = 1, \dots, n$, the cells of the decomposition can be divided into groups so that all cells of one group have the same $x_1, \dots, x_k$-coordinates. The precise definition is inductive:

- *one dimension:* Every algebraic decomposition of $\mathbb{R}$ is cylindrical.
- *n dimensions:* An algebraic decomposition of $\mathbb{R}^n$ is cylindrical if
  - the projection of any two cells down to $\mathbb{R}^{n-1}$ is either identical or disjoint, and
  - the projections of all the cells down to $\mathbb{R}^{n-1}$ form a cylindrical algebraic decomposition of $\mathbb{R}^{n-1}$.

The algebraic decomposition shown in Figure 1a is not cylindrical, because we can find pairs of cells (for example: the upper and the lower circle arc) which when projected to the horizontal axis have images that are neither identical nor disjoint. In order to make the decomposition cylindrical, we can refine it by adding a suitable univariate polynomial in x to P. Collins's algorithm finds this additional polynomial. Geometrically, adding a univariate polynomial means that we use vertical lines to split cells into smaller cells such as to satisfy the condition from the definition. In the example, a suitable polynomial is $(x^2-4)(x-1)(x^4-2x^3-2x^2+8x-4)$. The first factor corresponds to the vertical tangents of the circle, the second to the vertical asymptote of the hyperbola, and the third to the two intersection points of circle and hyperbola (Figure 1b).

Observe how the cells in Figure 1b are arranged into vertical stacks ("cylinders"): there are three cells stacked over the the range $-\infty < x < -4$, five with $x = -4$, seven with $-4 < x < \alpha$ where α is the leftmost real root of the polynomial $X^4 - 2X^3 - 2X^2 + 8X - 4$, and so on.

In the case of more than two variables, we may have to add more than one polynomial to make an algebraic decomposition cylindrical. For example, $P = \{x^2 + y^2 + z^2 - 1\}$ splits $\mathbb{R}^3$ into the interior, the boundary, and the exterior of the unit ball. This decomposition is not cylindrical. To make it cylindrical, we may first add $x^2 + y^2 - 1$ to P, which geometrically corresponds to putting a cylinder around the unit ball. Now every cell in the decomposition projects down to either the interior or the boundary or the exterior of the unit disk. But this is not enough. As the notion of being cylindrical applies recursively, we must also add $x^2 - 1$ to P corresponding to the two vertical tangents of the unit circle, or in the original picture, two vertical tangent planes. The resulting decomposition is then cylindrical (Figure 2).

One way of looking at Collins's algorithm is to say that it takes as input a finite set P of polynomials and produces as output another finite set Q of polynomials such that the algebraic decomposition induced by $P \cup Q$ is cylindrical. The output of the algorithm is rigorous in the sense that the decomposition of $P \cup Q$ is really cylindrical and not in any sense an approximation. A formal proof of the cylindricity could in principle be composed out of the intermediate expressions the CAD algorithm encounters during the computation, but checking such a proof would merely amount to redoing the whole calculation. The situation is therefore somewhat different from some summation algorithms which compute closed forms of symbolic sums along with a proof object (the so called "certificate") that can be easily checked independently of the computation. For CAD, you have to trust the program. Implementations that I consider trustworthy include *Mathematica*'s built-in, the free software QEPCAD [2], and *Redlog* [5].

4. The Logical Point of View

Individual cells in a cylindrical algebraic decomposition can be described by logical combinations of polynomial equations and inequalities. More generally, we may consider formulas which are constructed from variables, rational numbers, arithmetic operations ($+$, $-$, $\cdot$, $/$), equality and inequality relations ($=$, $\neq$, $<$, $>$, $\leq$, $\geq$), logical connectives ($\wedge$, $\vee$, $\dots$) and quantifiers ($\forall$, $\exists$) according to the usual syntactic rules.

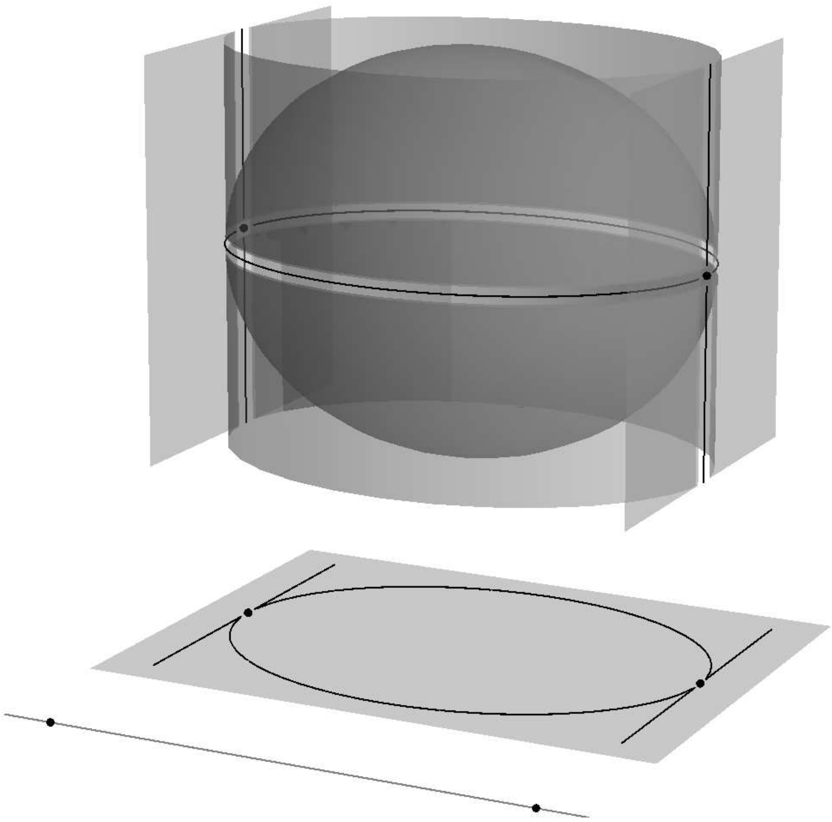

FIGURE 2. cylindrical algebraic decomposition of the unit ball

Some implementations allow some further functions to appear in formulas, for instance absolute values, minima and maxima, or algebraic functions. This is certainly convenient, but it does not make the theory more general, because formulas involving such functions can be easily rewritten into equivalent formulas which do not. For example,

$$0 \leq \max(x-y, x+y) \leq \sqrt{x^2+y^2}$$

can be rephrased as

$$\forall\, z : \Big(z \geq 0 \wedge z^2 = x^2+y^2\Big) \Rightarrow \Big(x-y \geq x+y \wedge 0 \leq x-y \leq z\Big) \vee \Big(x-y < x+y \wedge 0 \leq x+y \leq z\Big)$$

in a preprocessing step.

If formulas are interpreted in the theory of real numbers, then a formula without free variables is either true or false. A formula with k free variables can be regarded as a function $\mathbb{R}^k \to \{\text{true}, \text{false}\}$ which assigns to every point $(\xi_1, \ldots, \xi_k) \in \mathbb{R}^k$ the truth value of the formula obtained by replacing the free variables by the numbers $\xi_1, \ldots, \xi_k$. Two formulas are *equivalent* if they represent the same function. Note that although we use formulas to make statements about (sets of) real numbers, the formula themselves are not allowed to contain arbitrary real numbers but only rational ones.

Rephrased in terms of formulas, an alternative way of looking at Collins's algorithm is to say that it takes a formula as input and produces as output an equivalent formula which has a special structure. This special structure can be described recursively as follows:

- *one variable:* A formula in one variable x is in CAD format if it is of the form
$$\Phi_1 \vee \Phi_2 \vee \cdots \vee \Phi_m,$$
where each Φ_k is of the form $x < \alpha$ or $\alpha < x < \beta$ or $x > \beta$ or $x = \gamma$ for some real algebraic numbers α, β, γ ($\alpha < \beta$) and any two Φ_k are mutually inconsistent.
- *n variables:* A formula in n variables $x_1, \ldots, x_n$ is in CAD format if it is of the form
$$(\Phi_1 \wedge \Psi_1) \ \vee\ (\Phi_2 \wedge \Psi_2) \ \vee\ \cdots \ \vee\ (\Phi_m \wedge \Psi_m),$$
where the Φ_k are such that $\Phi_1 \vee \cdots \vee \Phi_m$ is in CAD format with respect to x_1 and the Ψ_k are satisfiable formulas which are in CAD format with respect to $x_2, \ldots, x_n$ whenever x_1 is replaced by a real algebraic number satisfying Φ_k.

Examples for formulas of this form have already been given in Section 2.

Note that the CAD format is defined in such a way that it naturally describes a union of some cells in a cylindrical algebraic decomposition of $\mathbb{R}^n$ as explained in the previous section. Once a formula has been brought to this format, the questions stated in the introduction become easy to answer. For example, in order to determine the dimension of a semialgebraic set, observe that a formula in CAD format can be expanded into a finite disjunction of formulas of the type
$$\Omega_1 \wedge \Omega_2 \wedge \cdots \wedge \Omega_n,$$
where each Ω_i is either an inequality $x_i < \alpha(x_1, \ldots, x_{i-1})$ or $\alpha(x_1, \ldots, x_{i-1}) < x_i < \beta(x_1, \ldots, x_{i-1})$ or $x_i > \beta(x_1, \ldots, x_{i-1})$ or an equality $x_i = \gamma(x_1, \ldots, x_{i-1})$ for some algebraic functions α, β, γ. Each of these conjunctions $\Omega_1 \wedge \Omega_2 \wedge \cdots \wedge \Omega_n$ describes an individual cell in the cylindrical decomposition of $\mathbb{R}^n$, and the dimension of that cell is obviously $n - E$ where E is the number of indices i for which Ω_i is not an inequality but an equality. For a semialgebraic set consisting of several cells, we simply determine the dimension for each cell in this way and take the maximum.

Quantifier elimination is also easy. Assume for example that we have a formula
$$(\Phi_1 \wedge \Psi_1) \ \vee\ (\Phi_2 \wedge \Psi_2) \ \vee\ \cdots \ \vee\ (\Phi_m \wedge \Psi_m)$$
in CAD format, say in two variables x_1, x_2. Then the quantified formula
$$\exists\, x_2 \in \mathbb{R} : (\Phi_1 \wedge \Psi_1) \ \vee\ (\Phi_2 \wedge \Psi_2) \ \vee\ \cdots \ \vee\ (\Phi_m \wedge \Psi_m)$$
is easily seen to be equivalent to the quantifier free formula
$$\Phi_1 \vee \Phi_2 \vee \cdots \vee \Phi_m$$
which only contains only x_1.

For the universal quantifier, we have to go through the Ψ_i and check which of them represent the whole real line, i.e., which of them are of the form

$$x_2 > \alpha \lor x_2 = \alpha \lor \alpha < x_2 < \beta \lor x_2 = \beta \lor \beta < x_2 < \gamma \lor x_2 = \gamma \lor \cdots \lor x_2 = \delta \lor x_2 > \delta.$$

(Good implementations will simplify such subformulas to True.) If it turns out that the relevant Ψ_i are Ψ_3, Ψ_7 and Ψ_{28}, for example, then the quantified formula

$$\forall\, x_2 \in \mathbb{R} : (\Phi_1 \land \Psi_1) \;\lor\; (\Phi_2 \land \Psi_2) \;\lor\; \cdots \;\lor\; (\Phi_m \land \Psi_m)$$

is equivalent to the quantifier free formula

$$\Phi_3 \lor \Phi_7 \lor \Phi_{28}.$$

Because of the recursive nature of formulas in CAD format, the case of more variables (and possibly more quantifiers) can be handled in quite the same way. The only issue to take care of is that the CAD format is with respect to a variable order which is compatible with the order of the quantifiers.

Quantifier elimination is the most important application of the CAD algorithm, and most implementations include elimination of quantifiers as a built-in feature, so that the user typically does not need to bother about choosing an appropriate variable order or constructing a quantifier free formula by picking the right parts out of a lengthy formula produced by the CAD algorithm. It is nevertheless useful to know what is going on behind the curtain, because it can give hints how to manually prepare the input so that a CAD computation can terminate more quickly.

5. Further Examples

Let us return to the inequality collection [9]. It contains many inequalities which can be checked by a simple CAD calculation.

5.1. Solving.

- Item 1.38 asks: For what values of x is it true that

  $$2 < (3x^2 - 15x + 16)/(x^2 - 4x + 3) < 3?$$

  Solution:

In[8]:= **CylindricalDecomposition[2 < (3x^2 − 15x + 16)/(x^2 − 4x + 3) < 3, {x}]**

Out[8]= $2 < x < \frac{7}{3} \lor x > 5$

- Item 1.51 asks: For which values of a does the following inequality hold:

  $$-1 < \frac{1}{2a}[1 - a - \sqrt{(1-a)^2 - 4a^2}] < +1?$$

  Solution:

In[9]:= **CylindricalDecomposition[−1 < 1/(2a)(1 − a − Sqrt[(1 − a)^2 − 4a^2]) < 1, {a}]**

Out[9]= $-1 < a < 0 \lor 0 < a < \frac{1}{3}$

- Item 1.8 asks: Solve the pair of inequalities
$$\frac{2x-y}{y}<0,\qquad \frac{2y-x}{x}<0\qquad (x,y\neq 0).$$
Solution:

In[10]:= **CylindricalDecomposition**$[\{x\neq 0, y\neq 0, \frac{2x-y}{y}<0, \frac{2y-x}{x}<0\}, \{x,y\}]$

Out[10]= $(x<0\wedge y>0)\vee(x>0\wedge y<0)$

- Item 1.29 asks: Determine the region of the xy-plane in which the point (x,y) must lie in order that its coordinates satisfy the inequality
$$(x^2-4xy)/(x^2+3xy+2y^2)<0.$$
Solution:

In[11]:= **CylindricalDecomposition**$[(x^2-4xy)/(x^2+3xy+2y^2)<0, \{x,y\}]$

Out[11]= $\left(x<0\wedge\left(y<\frac{1}{4}x\vee -\frac{1}{2}x<y<-x\right)\right)\vee\left(x>0\wedge\left(-x<y<-\frac{1}{2}x\vee y>\frac{1}{4}x\right)\right)$

- Item 1.39 asks: In a Cartesian coordinate system, find the region of the plane for which
$$1^\circ:\quad xy(x^2-y^2)>0,\qquad 2^\circ:\quad (x^2-1)(x^2-y^2)<0.$$
Solution:

In[12]:= **CylindricalDecomposition**$[\{xy(x^2-y^2)>0, (x^2-1)(x^2-y^2)<0\}, \{x,y\}]$

Out[12]= $(x<-1\wedge y>-x)\vee(-1<x<0\wedge x<y<0)\vee(0<x<1\wedge 0<y<x)\vee(x>1\wedge y<-x)$

- Item 1.49 asks: Find the region of the xy-plane for which the following inequalities are simultaneously satisfied:
$$x^2<7y,\ y^2>5x,\ y^2<8x,\ x^2>2y.$$
Solution:

In[13]:= **CylindricalDecomposition**$[\{x^2<7y,\ y^2>5x,\ y^2<8x,\ x^2>2y\}, \{x,y\}]$

Out[13]= $\left(\sqrt[3]{20}<x\leq\sqrt[3]{32}\wedge\sqrt{5x}<y<\frac{x^2}{2}\right)\vee\left(\sqrt[3]{32}<x\leq\sqrt[3]{245}\wedge\sqrt{5x}<y<2\sqrt{2x}\right)$
$\vee\left(\sqrt[3]{245}<x<\sqrt[3]{392}\wedge\frac{x^2}{7}<y<2\sqrt{2x}\right)$

- Item 2.29 asks: For what values of x is
$$(a-x)^6-3a(a-x)^5+\tfrac{5}{2}a^2(a-x)^4-\tfrac{1}{2}a^4(a-x)^2<0?$$
Solution:

In[14]:= **CylindricalDecomposition**[
$(a-x)^6-3a(a-x)^5+5/2a^2(a-x)^4-1/2a^4(a-x)^2<0, \{a,x\}]$

Out[14]= $\left(a<0\wedge(\tfrac{1}{2}(1+\sqrt{3})a<x<a\vee a<x<0\vee 0<x<\tfrac{1}{2}(1-\sqrt{3})a)\right)$
$\vee\left(a>0\wedge(\tfrac{1}{2}(1-\sqrt{3})a<x<0\vee 0<x<a\vee a<x<\tfrac{1}{2}(1+\sqrt{3})a)\right)$

Some manual clean-up of this output gives the equivalent answer

$$x \neq 0 \wedge x \neq a \wedge \frac{1}{2}(1-\sqrt{3}) < \frac{x}{a} < \frac{1}{2}(1+\sqrt{3}).$$

- Item 1.58 asks: Determine pairs of integers, x and y, which satisfy simultaneously

$$y - |x^2 - 2x| + \tfrac{1}{2} > 0, \quad y + |x-1| < 2.$$

Solution: First we determine all the real solutions via

In[15]:= $\mathbf{CylindricalDecomposition[\{y{-}Abs[x^2{-}2x]{+}\tfrac{1}{2} > 0,\ y{+}Abs[x{-}1] < 2\}, \{x, y\}]}$

Out[15]= $$\Big(\frac{1}{2}(3-\sqrt{15}) < x \leq 0 \wedge \frac{1}{2}(2x^2-4x-1) < y < x+1\Big)$$
$$\vee \Big(0 < x \leq 1 \wedge \frac{1}{2}(-2x^2+4x-1) < y < x+1\Big)$$
$$\vee \Big(1 < x \leq 2 \wedge \frac{1}{2}(-2x^2+4x-1) < y < 3-x\Big)$$
$$\vee \Big(2 < x < \frac{1}{2}(1+\sqrt{15}) \wedge \frac{1}{2}(2x^2-4x-1) < y < 3-x\Big)$$

This restricts the possible values for x to $0, 1, 2$. For these three particular values, the corresponding inequalities for y specialize as follows:

$$\begin{aligned} x = 0: &\quad -\tfrac{1}{2} < y < 1, \\ x = 1: &\quad \tfrac{1}{2} < y < 2, \\ x = 2: &\quad -\tfrac{1}{2} < y < 1. \end{aligned}$$

Therefore, the only integral solutions are $(0,0)$, $(1,1)$, and $(2,0)$.

5.2. **Proving.**

- Item 1.22 claims: If

$$f(a,b,c,d) = (a-b)^2 + (b-c)^2 + (c-d)^2 + (d-a)^2$$

then for a, b, c, d with $a < b < c < d$ we have

$$f(a,c,b,d) > f(a,b,c,d) > f(a,b,d,c).$$

Proof:

In[16]:= $\boldsymbol{f[a\_, b\_, c\_, d\_] = (a-b)^2 + (b-c)^2 + (c-d)^2 + (d-a)^2;}$

In[17]:= **CylindricalDecomposition[**
ForAll[{a, b, c, d}, a < b < c < d, f[a, c, b, d] > f[a, b, c, d] > f[a, b, d, c]], {}]

Out[17]= True

- Item 1.52 claims: If a and b ($ab \neq 0$) are arbitrary real numbers, then at least one of the following inequalities is valid:

$$\left|\frac{a+\sqrt{a^2+2b^2}}{2b}\right| < 1, \qquad \left|\frac{a-\sqrt{a^2+2b^2}}{2b}\right| < 1.$$

Proof:

In[18]:= **CylindricalDecomposition[**
$$\mathbf{ForAll[\{a,b\}, ab \neq 0, Abs[\tfrac{a+\sqrt{a^2+2b^2}}{2b}] < 1 \vee Abs[\tfrac{a-\sqrt{a^2+2b^2}}{2b}] < 1], \{\}]}$$
Out[18]= True

- Item 1.53 claims: $\left|\dfrac{x^2-2x+3}{x^2-4x+3}\right| \leq 1 \Rightarrow x \leq 0.$
  Proof:

In[19]:= **CylindricalDecomposition[Abs[$(x^2-2x+3)/(x^2-4x+3)$] $\leq 1, \{x\}$]**
Out[19]= $x \leq 0$

- Item 1.59 claims: $a^2+b^2+c^2 \geq |bc+ca+ab|$
  Proof:

In[20]:= **CylindricalDecomposition[ForAll[$\{a,b,c\}, a^2+b^2+c^2 \geq$ Abs[$bc+ca+ab$]], {}]**
Out[20]= True

- Item 1.60 claims: $a(a-b)(a-c)+b(b-c)(b-a)+c(c-a)(c-b) \geq 0$ $(a,b,c \geq 0)$.
  Proof:

In[21]:= **CylindricalDecomposition[**
ForAll[$\{a,b,c\}$, Min[a,b,c] ≥ 0,
$a(a-b)(a-c)+b(b-c)(b-a)+c(c-a)(c-b) \geq 0$], {}]
Out[21]= True

- Item 1.61 claims: $\frac{ab}{a+b}+\frac{cd}{c+d} \leq \frac{(a+c)(b+d)}{a+b+c+d}$ $(a,b,c,d>0)$.
  Proof:

In[22]:= **CylindricalDecomposition[**
$$\mathbf{ForAll[\{a,b,c,d\}, Min[a,b,c,d] > 0, \frac{ab}{a+b}+\frac{cd}{c+d} \leq \frac{(a+c)(b+d)}{a+b+c+d}], \{\}]}$$
Out[22]= True

- Item 1.7 claims: If a,b,c,d are real numbers and if $ad-bc=1$, then $a^2+b^2+c^2+d^2+ac+bd>1$.
  Proof:

In[23]:= **CylindricalDecomposition[**
ForAll[$\{a,b,c,d\}, ad-bc == 1, a^2+b^2+c^2+d^2+ac+bd > 1$], {}]
Out[23]= True

The bound 1 on the right hand side is not sharp. To find the best bound, redo the computation with a symbolic bound M.

In[24]:= **CylindricalDecomposition[**
ForAll[$\{a,b,c,d\}, ad-bc == 1, a^2+b^2+c^2+d^2+ac+bd > M$], $\{M\}$]
Out[24]= $M < \sqrt{3}$

This is essentially the statement of Item 1.23 and an example for quantifier elimination.

5.3. Quantifier Elimination.

- Item 1.30 asks: Find the region of the plan in a Cartesian coordinates system whose points (x, y) satisfy the condition

$$\big||x+a| - |y-a|\big| < a \qquad (a > 0).$$

Solution:

In[25]:= $\mathbf{CylindricalDecomposition[ForAll[}a, a > 0, \mathbf{Abs[Abs[}x+a\mathbf{]} - \mathbf{Abs[}y-a\mathbf{]]} < a\mathbf{], \{}x, y\mathbf{\}]}$

Out[25]= $x == -y$

- Item 1.34 asks: For what value or values of a is the condition

$$(x^2+ax+1)/(x^2+4x+8) < 8$$

satisfied for all real x?
Solution:

In[26]:= $\mathbf{CylindricalDecomposition[ForAll[}x, (x^2+ax+1)/(x^2+4x+8) < 8\mathbf{], \{}a\mathbf{\}]}$

Out[26]= $-10 < a < 74$

- Item 1.35 asks: Determine k such that for all real x

$$|(x^2-kx+1)/(x^2+x+1)| < 3.$$

Solution:

In[27]:= $\mathbf{CylindricalDecomposition[ForAll[}x, \mathbf{Abs[}(x^2-kx+1)/(x^2+x+1)\mathbf{]} < 3\mathbf{], \{}k\mathbf{\}]}$

Out[27]= $-5 < k < 1$

- Item 1.53 claims: $\frac{p+m}{p+m} \geq \frac{x^2-2mx+p^2}{x^2+2mx+p^2} \geq \frac{p-m}{p+m}$ $(p > m > 0)$
Solution: Let us drop the condition $p > m > 0$ and let CAD determine the region in the pm-plane where the inequality holds.

In[28]:= **CylindricalDecomposition[**

$$\mathbf{ForAll[}x, \frac{p+m}{p+m} \geq \frac{x^2-2mx+p^2}{x^2+2mx+p^2} \geq \frac{p-m}{p+m}\mathbf{], \{}p, m\mathbf{\}]}$$

Out[28]= $(p < 0 \wedge p < m <= 0) \vee (p > 0 \wedge 0 <= m < p)$

- Item 7.5 claims: If the sum of four positive numbers is $4c$ and the sum of their squares is $8c^2$, then none of the numbers can exceed $(1+\sqrt{3})c$.
Proof: Let a_1, a_2, a_3, a_4 be the four positive numbers. Without loss of generality, $a_1 \geq a_2 \geq a_3 \geq a_4$. Then it suffices to prove the claim for a_1.

In[29]:= **CylindricalDecomposition[**

$$\mathbf{Exists[\{}a_2, a_3, a_4\mathbf{\}}, a_1 \geq a_2 \geq a_3 \geq a_4 > 0 \wedge a_1+a_2+a_3+a_4 == 4c \wedge a_1^2+a_2^2+a_3^2+a_4^2 == 8c^2\mathbf{]}, c, a_1\mathbf{]}$$

Out[29]= $c > 0 \wedge 2c < a_1 \leq (1+\sqrt{3})c$

This implies in particular that the bound is sharp.

- Item 11.15 asks: Which conditions must be satisfied by the coefficients a, b, c, d, e, f for the function

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f$$

to be positive for all real values of x and y?
Solution:

In[30]:= **CylindricalDecomposition[**
ForAll[{x, y}, $ax^2 + 2bxy + cy^2 + 2dx + 2ey + f > 0$], {$a, b, c, d, e, f$}]

produces an answer within a few minutes. It is a bit lengthy (some three pages), so we do not reprint it here. Let us instead just show the output for the case when a, b, c are restricted to the positive numbers.

In[31]:= **CylindricalDecomposition[**
ForAll[{x, y}, Min[a, b, c] > 0
$\wedge\, ax^2 + 2bxy + cy^2 + 2dx + 2ey + f > 0$], {$a, b, c, d, e, f$}]

Out[31]= $a > 0 \wedge b > 0 \wedge \Big(\Big(c = \frac{b^2}{a} \wedge e = \frac{cd}{b} \wedge f > \frac{de}{b}\Big) \vee \Big(c > \frac{b^2}{a} \wedge \Big(\Big(d < 0 \wedge f > \frac{ae^2 - 2bde + cd^2}{ac - b^2}\Big) \vee \Big(d = 0 \wedge \Big(\Big(e < 0 \wedge f > \frac{ae^2}{ac - b^2}\Big) \vee (e = 0 \wedge f > 0) \vee \Big(e > 0 \wedge f > \frac{ae^2}{ac - b^2}\Big)\Big)\Big) \vee \Big(d > 0 \wedge f > \frac{ae^2 - 2bde + cd^2}{ac - b^2}\Big)\Big)\Big)\Big)$

- Item 1.57 asks: Find lower and upper bounds for the function

$$(x^2 - 2x \cos a + 1)/(x^2 - 2x \cos b + 1).$$

Solution: First we have to get rid of the trigonometric functions because CAD can only deal with polynomial expressions. In the present case this is easily done by replacing $\cos a$ and $\cos b$ by new variables A and B, respectively, which are constrained to the interval $[-1, 1]$. Then

In[32]:= **CylindricalDecomposition[**
ForAll[x, Abs[A] $\leq 1 \wedge$ Abs[B] ≤ 1, $L \leq \frac{x^2-2xA+1}{x^2-2xB+1} \leq U$], {$A, B, L, U$}]

Out[32]= $A < -1 \vee \Big(A = -1 \wedge \Big(B < -1 \vee \Big(-1 < B < 1 \wedge L \leq 0 \wedge U \geq -\frac{2}{B-1}\Big) \vee B > 1\Big)\Big) \vee \Big(-1 < A < 1 \wedge \Big(B < -1 \vee \Big(-1 < B < A \wedge L \leq \frac{A-1}{B-1} \wedge U \geq \frac{A+1}{B+1}\Big) \vee (B = A \wedge L \leq 1 \wedge U \geq 1) \vee \Big(A < B < 1 \wedge L \leq \frac{A+1}{B+1} \wedge U \geq \frac{A-1}{B-1}\Big) \vee B > 1\Big)\Big) \vee \Big(A = 1 \wedge \Big(B < -1 \vee \Big(-1 < B < 1 \wedge L \leq 0 \wedge U \geq \frac{2}{B+1}\Big) \vee B > 1\Big)\Big) \vee A > 1$

implies the following sharp upper and lower bounds in terms of A and B:

$$\begin{array}{lll} -1 < A \leq B < 1: & L = \dfrac{1+A}{1+B} & U = \dfrac{1-A}{1-B} \\ -1 < B \leq A < 1: & L = \dfrac{1-A}{1-B} & U = \dfrac{1+A}{1+B} \\ |A| = 1 \wedge |B| \neq 1: & L = 0 & U = \dfrac{2}{1+AB}. \end{array}$$

Bounds which are independent of A and B cannot be given because

In[33]:= **CylindricalDecomposition[**
ForAll[{x, A, B}, Abs[A] $\leq$ 1 $\wedge$ Abs[B] $\leq$ 1, $L \leq \frac{x^2-2xA+1}{x^2-2xB+1} \leq U$], {$L, U$}]

Out[33]= False

6. Further Remarks

Our example collection hopefully sustains the claim that the CAD algorithm is useful. It was not unintentional that the examples we have chosen are simple enough that they could as well be done by hand with a reasonable effort. The point is that even in such situations CAD is is more reliable, gives best-possible answers, saves working time, and shortens proofs compared to traditional paper and pencil reasoning.

Sometimes it is possible to rephrase a problem as a formula with polynomial inequalities which is so involved that it seems completely hopeless to prove it by hand. In such situations, even if CAD is applicable in principle, it is often not successful in practice, because the runtime and memory requirements of the CAD algorithm can easily become too astronomic to get a computation done. The question is then whether the problem at hand can be broken by paper and pencil reasoning into smaller pieces which can then be solved by CAD in a reasonable amount of time. An example can be found in [8].

References

[1] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer, 2nd edition, 2006.

[2] Chris W. Brown. QEPCAD B – a program for computing with semi-algebraic sets. *Sigsam Bulletin*, 37(4):97–108, 2003.

[3] Bob F. Caviness and Jeremy R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation. Springer, 1998.

[4] George E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. *Lecture Notes in Computer Science*, 33:134–183, 1975.

[5] Andreas Dolzmann and Thomas Sturm. Redlog: computer algebra meets computer logic. *Sigsam Bulletin*, 31(2):2–9, 1997.

[6] Manuel Kauers. Computer algebra and special function inequalities. In Tewodros Amdeberhan and Victor H. Moll, editors, *Tapas in Experimental Mathematics*, volume 457 of *Contemporary Mathematics*, pages 215–235. AMS, 2008.

[7] Manuel Kauers and Veronika Pillwein. When can we decide that a P-finite sequence is positive? In *Proceedings of ISSAC'10*, 2010. 195–202.

[8] Manuel Kauers, Veronika Pillwein, and Susanne Saminger-Platz. Dominance in the family of Sugeno–Weber t-norms. Technical Report, 2010, `arχiv:1007.5442`.
[9] Dragoslav S. Mitrinović. *Elementary Inequalities*. P. Noordhoff Ltd., 1964.
[10] Veronika Pillwein. Positivity of certain sums over Jacobi kernel polynomials. *Advances in Applied Mathematics*, 41(3):365–377, 2008.
[11] Adam Strzeboński. Solving systems of strict polynomial inequalities. *Journal of Symbolic Computation*, 29:471–480, 2000.

Manuel Kauers, Research Institute for Symbolic Computation, J. Kepler University Linz, Austria

E-mail address: `mkauers@risc.uni-linz.ac.at`